Markdown+Pandoc: A Swiss Army Knife for Content Creation

OpenLearningWV Statewide Convening

Robert Szarka

Shepherd University

2024-04-05

Introduction

Preparation: Set Up StackEdit

If you want to play along while I demonstrate the basics of Markdown later in this talk, go to StackEdit.io.

If you also authorize StackEdit to use your Google account, it will sync your files via Google Drive.

Who am I?

Rob Szarka:

- Assistant Professor of Economics, Shepherd University
- Sysadmin hosting Internet services since 1992
- Bitcoiner
- Swing dancer

What am I doing?

Writing a Principles of Economics text and producing it in multiple formats:

- PDF for offline reading & printing
- HTML for reading via the LMS
- EPUB for e-book readers
- Beamer PDF & PowerPoint for slides

How am I doing it?

- Markdown: “an easy-to-read, easy-to-write plain text format”
- Pandoc: “a universal document converter”

What do I hope to accomplish today?

I would like to convince you that:

- Pandoc is a useful tool that can solve document conversion problems.
- Markdown is a useful format that can fit into your workflow in many places, even if you don't try to write an entire textbook in it.
- open source software & open formats are the logical way to create open content.

Pandoc

What is Pandoc?

- free (as in beer **&** freedom) software licensed under the GPL
- written by John MacFarlane, Professor of Philosopy at UC Berkeley
- "a universal document converter"

Example: LaTeX to MS Word

My original use case: convert my LaTeX document to MS Word for publication:

```
pandoc -s -o RTSpaper.docx -f latex -t docx RTSpaper.tex
```

Example: MS Word to ODF

Let's convert that filthy Microsoft format to Open Document Format for use with LibreOffice/OpenOffice:

```
pandoc -o RTSpaper.odt RTSpaper.docx
```

Example: MS Word to ODF (continued)

Couldn't I just do that with MS Word?

- Not if you don't own Word!
- Can Word do it in an automated fashion?

Pandoc can easily be called from a script, or used as part of a toolchain (can use STDIN for input and/or STDOUT for output).

Example: PubPub

As an example of the kind of automation that's possible with Pandoc, consider the open-source publishing platform PubPub. (I learned about PubPub yesterday from Jeremy Larance, who curates the British Literature OER site.)

Judging from their Github repository, PubPub relies on Pandoc to power at least some of their document conversions.

Example: One Weird Trick for Scraping the Web

```
pandoc -s -r html https://szarka.org/ -o szarka.md
```

This is similar to `Save to PDF` in your browser or downloading with wget, but

- it's scriptable
- it can save to *any* format Pandoc supports

tl;dr Pandoc Is a Beast!

- If you ever run into a “document” you can’t convert, Pandoc has your back.
- If you have a whole mess of documents to convert, ask you instructional designer to script the conversion process with Pandoc?
- Please buy John MacFarlane a beer for me!

Markdown

Origins of Markdown

Created in 2004 by Aaron Swartz & John Gruber of Daring Fireball:

> *Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML). . . . The overriding design goal for Markdown's formatting syntax is to make it as readable as possible.*

Origins of Markdown (continued)

> *While Markdown's syntax has been influenced by several existing text-to-HTML filters, the single biggest source of inspiration for Markdown's syntax is the format of plain text email. The best way to get a feel for Markdown's formatting syntax is simply to look at a Markdown-formatted document.*

OK, let's look at the Markdown that made these slides. */me fumbles with his laptop*

Slide Notes Example

This slide has notes in PowerPoint.

Markdown's Philosophy

Gruber's original software for converting Markdown to HTML was written in Perl, and Markdown itself shares some of Perl's ethos:

- less designed and more evolved from prior practice
- TIMTOWTDI (There's More Than One Way To Do It)

Markdown's Philosophy (continued)

As might be expected, the original Markdown specification has since spawned several dialects of Markdown. (Although there have also been some attempts at standardization—or, at least, documentation of existing practice—in RFC 7763 (The text/markdown Media Type) and RFC 7764 (Guidance on Markdown).)

Markdown vs. HTML

- Unlike, e.g., MS Word's .docx files, a Markdown file is just "plain" text.
- Unlike, e.g., HTML, Markdown approximates a "natural" way of writing, and it's readable even in its untranslated form.
- Markdown is similar in spirit (& sometimes syntax) to other lightweight markup languages, e.g., MediaWiki or BBCode.

Markdown vs. HTML: Example 1

Markdown

```
This---this!---is an **important** paragraph
with a [link](someurl.html "some page").
```

HTML

```
<p>This&mdash;this!&mdash;is an
<strong>important</strong> paragraph
with a <a href="someurl" title="some page">link</a>.</p>
```

Markdown vs. HTML: Example 2

Markdown

```
- list item 1
  - second level list item
```

HTML

```
<ul>
<li>list item 1
  <ul>
  <li>second level list item</li>
  </ul>
</li>
</ul>
```

Markdown vs. HTML: Example 3

Markdown

```
$A = \pi r^2$
```

HTML (MathML, best case)

```
<math><mrow>
  <mi>A</mi><mo>=</mo><mi>&pi;</mi>
    <msup>
      <mrow><mi>r</mi></mrow>
      <mrow><mn>2</mn></mrow>
    </msup>
</mrow></math>
```

Markdown Syntax

To explore Markdown syntax further, see the Markdown Cheat Sheet in StackEdit or try MarkdownGuide.org.

Markdown in Your Daily Life

Adopting Markdown doesn't have to be an all or nothing decision!

- Use basic Markdown commands to format text as you type in Google Docs and on many web sites.
- Jot down quick notes in a text editor like Notepad or an app like Google Keep.
- Use a WYSIWYG tool like StackEdit or Macdown to quickly draft documents, then export to another format to revise.
- Use Markdown in dedicated note taking apps like Obsidian or Joplin

My Workflow

Write Markdown

- Optionally draft a chapter in StackEdit or pull notes from Obsidian as a starting point
- Revise using vi (or vim)

Produce HTML for Individual Chapters

```
pandoc -f markdown -t html5 "chapter.md" -o "chapter.html"
--ascii --webtex --embed-resources
```

Produce HTML for Individual Chapters (continued)

- `--ascii` avoids character set issues
- `--webtex` uses an online service to create graphics for equations
- `--embed-resources` embeds graphics in the file, saving the work of uploading graphics to the LMS separately (big time saver!)

Paste HTML into the LMS

- By default, Pandoc creates an HTML fragment, perfect for pasting in. (The `--standalone` option gives you a complete document, if you need one.)
- I've done this successfully with Blackboard, Canvas, & Brightspace. Equations is the tricky part, but there are at least three different ways to do it!

Compile a PDF File from All the Chapters

Pandoc will read in multiple files and compile them into one document, with options (e.g., columns, running headers, margins optimized for print, etc.) given on the command line or in a YAML block at the beginning.

If we have time, let's look at my book draft as an example.

Compile an EPUB E-Book File from All the Chapters

Very similar workflow to creating the PDF, but an EPUB file is basically HTML under the hood. Some new options to tweak, e.g., cover image.

Advanced Stuff I'm Working On

- cross-references
- integrate with BibTeX for citations
- use reference docs (custom templates) to tweak formatting
- automate workflow with make
- IMS Content Packaging

Open Source

Open Access Started With Open Source

The movement toward Open Culture, Open Access, OER, etc., started with Open Source / Free Software. And OSS has its roots in Unix. Unix and OSS have *their* roots in academia and research labs (AT&T). Creating OER content and OSS is a perfect match.

Today's Sermon

Universities shouldn't be paying to lock up knowledge in proprietary silos, dependent on closed source software and proprietary formats. We can control the tools we use to author content, but we need to demand more from our institutions and our vendors, too.

The End?

Thank You

- Email me at `rszarka@shepherd.edu` for slides.
- Maybe we should have an OER tech mailing list?
- Questions?