

2MB + 2MB = 22MB

Sep 18, 2024 • spencerburleigh.com/tidydata

Preview on a Mac will merge two PDFs for you. It will also, in my experience, turn a pair of 2MB files into a 22MB one. Notion will export a [second brain](#)^[a] as a zip of folders named with 32-character hashes. A phone photo posted to a website is likely to carry timestamps and GPS coordinates unless you strip the metadata.

I got tired of solving each of those once and throwing the script away. [tidyData](#)^[b] is the drawer I keep them in.

Don't let Preview merge your PDFs

`combinePDFS.py` uses [pypdf](#)^[c] to copy pages, then checks whether the output is larger than the sum of the inputs and prints a warning if it is.

There's a sibling for markdown (`combineMarkdown.py` concatenates every `.md` in a folder with `# filename` as the delimiter) and one for a grab-bag of text-ish extensions like plain-text transcripts. This has been really helpful for managing how I give context to LLMs. I use the markdown one after a Notion export, when I want one file I can search instead of forty.

Notion dumps are a tree. I want a pile.

`surfaceFiles.py` is the other half of that workflow. It walks an export, removes long hash strings from names, and copies selected attachment types into a flatter folder structure. The source files stay in place, and files whose cleaned names collide get numbered suffixes so neither copy is overwritten.

`folderSizer.py` is the “what is taking up space” version of the same instinct: print a tree with file-type counts and make a text report to help me see what's worth cleaning up before I move things around.

Photos without metadata

When I created the [Photos](#)^[d] section of this website, I wanted to maintain some amount of privacy around timestamps and location. The metadata-stripping function in `imagekit.py` applies EXIF rotation to the pixels, then copies those pixels into a fresh image without carrying over the source

metadata, including GPS coordinates, camera details, and timestamps. It can also generate thumbnails at a width you choose.

There's an optional S3 integration if you want to process images stored in a bucket the way I do for this site.

The rest of the drawer

`epubToMarkdown.py` for books I want as text. `docSplice.py` for splitting a long Word doc at two marker strings.

Python 3.9+, MIT. The image tools can be installed as a library; the repository also includes the standalone file utilities.

```
python -m pip install "git+https://github.com/sburl/tidyData.git"
# For S3 support, use this instead:
python -m pip install "tidydata[s3] @ git+https://github.com/sburl/tidyData.git"
```

Links

- a. spencerburleigh.com/founders-guide-to-second-brain
- b. github.com/sburl/tidyData
- c. pypi.org/project/pypdf
- d. spencerburleigh.com/photos