---

title: Tufte Pandoc CSS subtitle: Starter files for Pandoc Markdown with Tufte CSS author: Jake Zimmerman

date: ’November 3, 2016’

Tufte Pandoc CSS is an attempt to make it as easy as possible to get started using [Tufte CSS][^tufte-css] to write content. It does this by leveraging Pandoc Markdown’s existing features, along with a few new ones implemented as a JSON filter.

[^tufte-css]: {-} If you’ve never heard of Tufte CSS before, take a second to check it out!

Sidenotes in Markdown

```
<div class="epigraph">
> Tufte CSS provides tools to style web articles using the ideas demonstrated by
> Edward Tufte's books and handouts. Tufte's style is known for its simplicity,
> extensive use of sidenotes, tight integration of graphics with text, and
> carefully chosen typography.
>
> <footer>
> Dave Liepmann, [Tufte CSS]
> </footer>
</div>
```

Tufte Pandoc CSS aims to be a set of *starter files* for your next project. What that means is that it provides a number of CSS files, a Pandoc template, a Makefile, and more to make it as easy as possible to write Markdown using Tufte CSS.

The biggest barrier that this project overcomes is that Pandoc Markdown doesn’t support side notes nor margin notes by default. This project adds that functionality.[^pdsn] Here’s how you can use them:

[^pdsn]: In particular, a separate library called `pandoc-sidenote` handles the transformation of footnotes into sidenotes.

“‘markdown ... In print, Tufte has used the proprietary Monotype Bembo[^note] font. ...

[^note]: See Tufte’s comment in the Tufte book fonts thread. “‘

By default, if you use the normal Pandoc syntax for creating footnotes, they’ll become Tufte CSS-style side notes. To get margin notes (i.e., side notes without numbers), just include `{-}` at the beginning of the note:

`<figure class="fullwidth">` “‘markdown ... If you want a sidenote without footnote-style numberings, then you want a margin note.[^mn] On large screens, ...

[^mn]: {-} This is a margin note. Notice there isn’t a number preceding the note. “‘ </figure>

Syntax-Highlighted Codeblocks

Pandoc Markdown is excellent for styling code blocks. Indeed, you’ve already seen their effect in this document. You can use any of the methods for creating syntax highlighted code blocks Pandoc permits. For example:

~~~python
~~~

Compute elements of the mandelbrot set

def mandelbrot(a): return reduce(lambda z, _: z * z + a, range(50), 0)

def step(start, step, iterations): return (start + (i * step) for i in range(iterations))

rows = ((”*” if abs(mandelbrot(complex(x, y))) < 2 else ” ” for x in step(-2.0, .0315, 80)) for y in step(1, -.05, 41)) ~~~

In this document, you’re also seeing the effect of the Solarized color scheme. You can use any of the color schemes Pandoc ships with by default, or this one.

Another feature Pandoc allows that Tufte Pandoc CSS supports is generating line numbers to accompany a code sample:

“‘{.haskell .numberLines} merge [] ys = ys merge xs [] = xs merge xs@(x:xt) ys@(y:yt) | x <= y = x : merge xt ys | otherwise = y : merge xs yt

split (x:y:zs) = let (xs,ys) = split zs in (x:xs,y:ys) split [x] = ([x],[]) split [] = ([],[])

mergeSort [] = [] mergeSort [x] = [x] mergeSort xs = let (as,bs) = split xs in merge (mergeSort as) (mergeSort bs) “‘

Figures & Sections

Unfortunately, HTML figures and sections don’t have special Markdown syntax. To include them in your document, you’ll have to use the raw HTML. On the bright side, this usually ends up being pretty painless.

In particular for sections, if you’re satisfied with the top-most headings being wrapped in `<section>` tags, you can use the `--section-divs` flag to `pandoc` to automatically wrap sections in divs. This is already enabled in the Makefile we ship with this project.

Tufte Pandoc CSS improves support for full-width tables and code blocks. Special attention has been given to ensure that they’re fully responsive at all viewports, just like normal full-width figures. Here’s a sample full-width table:

`<figure class="fullwidth">`

| Talk | Speaker | Time | | —- | ——- | —- | | HTML/CSS Primer | Scott Krulcik | 1:30 p.m. – 3:00 p.m. | | JavaScript Primer | Jake Zimmerman | 3:00 p.m. – 4:30 p.m. | | UX Prototyping with Framer.js | Lois Yang | 4:30 p.m. – 6:00 p.m. | | Frontend Development with Angular.js | Sandra Sajeev | 6:30 p.m. – 8:00 p.m. |

```
</figure>
```

As one last quick note: the original Tufte CSS recommends that you always wrap images in `<figure>` tags for optimal responsiveness and layout. Depending on your tastes, you **can** choose to omit this. The differences will only take effect on mobile, where the width of the image will be slightly different from what it would be if it were properly wrapped. Try it both ways and see whether you value the convenience of no wrapping or the proper layout that a `<figure>` provides.

Installation & Usage

As mentioned above, Tufte Pandoc CSS is designed to be a collection of starter files to help you with your next Markdown project. You can learn what files and tools are available on the GitHub repository.

As for usage, you are strongly encouraged to look at the source of this document. There's also an HTML-to-Markdown port of the original Tufte CSS page along with the accompanying source.

One goal of this project is to support as many of the features you'd ”expect” to work that are available in Pandoc. If your favorite feature doesn't work, let us know with an issue.