Tufte Handout Sample

David Peng

September 25, 2019

This sample text is from Dave Liepmann's Tufte CSS.

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.

Tufte CSS was created by Dave Liepmann and is now an Edward Tufte project. The original idea was cribbed from Tufte-LaTeX and R Markdown's Tufte Handout format. We give hearty thanks to all the people who have contributed to those projects.

If you see anything that Tufte CSS could improve, we welcome your contribution in the form of an issue or pull request on the GitHub project: tufte-css. Please note the contribution guidelines.

Finally, a reminder about the goal of this project. The web is not print. Webpages are not books. Therefore, the goal of Tufte CSS is not to say "websites should look like this interpretation of Tufte's books" but rather "here are some techniques Tufte developed that we've found useful in print; maybe you can find a way to make them useful on the web". Tufte CSS is merely a sketch of one way to implement this particular set of ideas. It should be a starting point, not a design goal, because any project should present their information as best suits their particular circumstances.

Getting Started

To use Tufte CSS, copy `tufte.css` and the `et-book` directory of font files to your project directory, then add the following to your HTML document's `head` block:

```
<link rel="stylesheet" href="tufte.css"/>
```

Now you just have to use the provided CSS rules, and the Tufte CSS conventions described in this document. For best results, View Source and Inspect Element frequently.

Fundamentals

Sections and Headings

Organize your document with an `article` element inside your `body` tag. Inside that, use `section` tags around each logical grouping of text and headings.

Tufte CSS uses `h1` for the document title, `p` with class `subtitle` for the document subtitle, `h2` for section headings, and `h3` for low-level headings. More specific headings are not supported. If you feel the urge to reach for a heading of level 4 or greater, consider redesigning your document.

As a bonus, this excerpt regarding the use of headings provides an example of block quotes. In Tufte CSS they are just lightly styled, semantically correct HTML using `blockquote` and `footer` elements. See page 20 of The Visual Display of Quantitative Information for an example in print.

In his later books[1], Tufte starts each section with a bit of vertical space, a non-indented paragraph, and the first few words of the sentence set in small caps. For this we use a span with the class `newthought`, as demonstrated at the beginning of this paragraph. Vertical spacing is accomplished separately through `<section>` tags. Be consistent: though we do so in this paragraph for the purpose of demonstration, do not alternate use of header elements and the `newthought` technique. Pick one approach and stick to it.

[1] *Beautiful Evidence*

Text

Although paper handouts obviously have a pure white background, the web is better served by the use of slightly off-white and off-black colors. Tufte CSS uses `#fffff8` and `#111111` because they are nearly indistinguishable from their 'pure' cousins, but dial down the harsh contrast. We stick to the greyscale for text, reserving color for specific, careful use in figures and images.

In print, Tufte has used the proprietary Monotype Bembo [2] font. A similar effect is achieved in digital formats with the now open-source ETBook, which Tufte CSS supplies with a `@font-face` reference to a .ttf file. In case ETBook somehow doesn't work, Tufte CSS shifts gracefully to other serif fonts like Palatino and Georgia.

[2] See Tufte's comment in the Tufte book fonts thread.

Also notice how Tufte CSS includes separate font files for bold (strong) and italic (emphasis), instead of relying on the browser to mechanically transform the text. This is typographic best practice.

If you prefer sans-serifs, use the `sans` class. It relies on Gill Sans, Tufte's sans-serif font of choice.

Links in Tufte CSS match the body text in color and do not change on mouseover or when clicked. Here is a dummy example that goes nowhere. These links are underlined, since this is the most widely recognized indicator of clickable text. However, because most browsers' default underlining does not clear descenders and is so thick and distracting, the underline effect is instead achieved using CSS trickery

Blue text, while also a widely recognizable clickable-text indicator, is crass and distracting. Luckily, it is also rendered unnecessary by the use of underlining.

involving background gradients instead of standard `text-decoration`. Credit goes to Adam Schwartz for that technique.

As always, these design choices are merely one approach that Tufte CSS provides by default. Other approaches, such as changing color on click or mouseover, or using highlighting or color instead of underlining to denote links, could also be made to work. The goal is to make sentences readable without interference from links, as well as to make links immediately identifiable even by casual web users.

Sidenotes: Footnotes and Marginal Notes

One of the most distinctive features of Tufte's style is his extensive use of sidenotes.[3] Sidenotes are like footnotes, except they don't force the reader to jump their eye to the bottom of the page, but instead display off to the side in the margin. Perhaps you have noticed their use in this document already. You are very astute.

[3] This is a sidenote.

Sidenotes are a great example of the web not being like print. On sufficiently large viewports, Tufte CSS uses the margin for sidenotes, margin notes, and small figures. On smaller viewports, elements that would go in the margin are hidden until the user toggles them into view. The goal is to present related but not necessary information such as asides or citations *as close as possible* to the text that references them. At the same time, this secondary information should stay out of the way of the eye, not interfering with the progression of ideas in the main text.

Sidenotes consist of two elements: a superscript reference number that goes inline with the text, and a sidenote with content. To add the former, just put a label and dummy checkbox into the text where you want the reference to go, like so:

```
<label for="sn-demo" class="margin-toggle sidenote-number"></label>
<input type="checkbox" id="sn-demo" class="margin-toggle"/>
```

You must manually assign a reference `id` to each side or margin note, replacing "sn-demo" in the `for` and the `id` attribute values with an appropriate descriptor. It is useful to use prefixes like `sn-` for sidenotes and `mn-` for margin notes.

Immediately adjacent to that sidenote reference in the main text goes the sidenote content itself, in a `span` with class `sidenote`. This tag is also inserted directly in the middle of the body text, but is either pushed into the margin or hidden by default. Make sure to position your sidenotes correctly by keeping the sidenote-number label close to the sidenote itself.

If you want a sidenote without footnote-style numberings, then you want a margin note. On large screens, a margin note is just a

This is a margin note. Notice there isn't a number preceding the note.

sidenote that omits the reference number. This lessens the distracting effect taking away from the flow of the main text, but can increase the cognitive load of matching a margin note to its referent text. However, on small screens, a margin note is like a sidenote except its viewability-toggle is a symbol rather than a reference number. This document currently uses the symbol (`&#8853;`), but it's up to you.

Margin notes are created just like sidenotes, but with the `marginnote` class for the content and the `margin-toggle` class for the label and dummy checkbox. For instance, here is the code for the margin note used in the previous paragraph:

```
<label for="mn-demo" class="margin-toggle">&#8853;</label>
<input type="checkbox" id="mn-demo" class="margin-toggle"/>
<span class="marginnote">
    This is a margin note. Notice there isn't a number preceding the note.
</span>
```

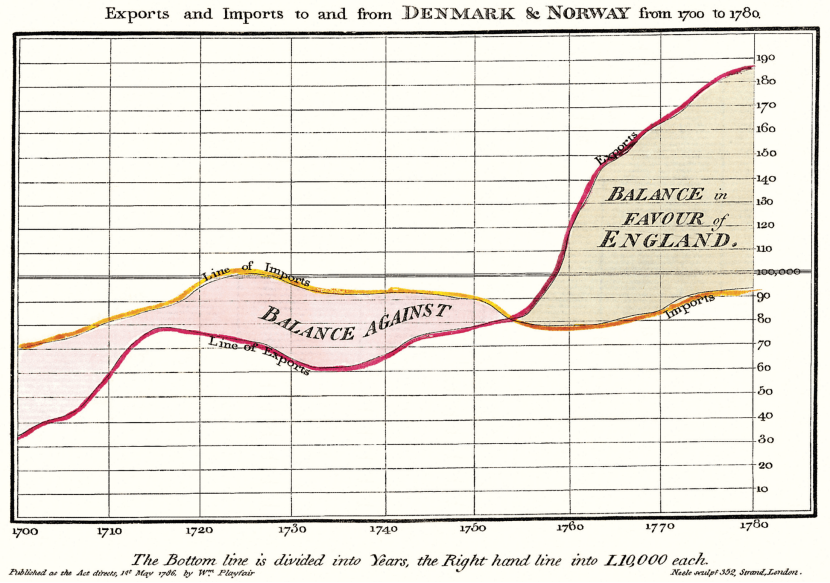
Figures in the margin are created as margin notes, as demonstrated in the next section.

Figures

Tufte emphasizes tight integration of graphics with text. Data, graphs, and figures are kept with the text that discusses them. In print, this means they are not relegated to a separate page. On the web, that means readability of graphics and their accompanying text without extra clicks, tab-switching, or scrolling.

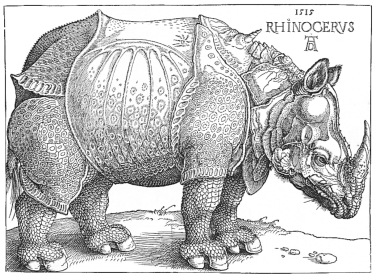
Figures should try to use the `figure` element, which by default are constrained to the main column. Don't wrap figures in a paragraph tag. Any label or margin note goes in a regular margin note inside the figure. For example, most of the time one should introduce a figure directly into the main flow of discussion, like so:

From Edward Tufte, *Visual Display of Quantitative Information*, page 92.



But tight integration of graphics with text is central to Tufte's work even when those graphics are ancillary to the main body of a text. In many of those cases, a margin figure may be most appropriate. To place figures in the margin, just wrap an image (or whatever) in a margin note inside a `p` tag, as seen to the right of this paragraph.

If you need a full-width figure, give it the `fullwidth` class. Make sure that's inside an `article`, and it will take up (almost) the full width of the screen. This approach is demonstrated below using Edward Tufte's English translation of the Napoleon's March data visualization. From *Beautiful Evidence*, page 122-124.



F.J. Cole, "The History of Albrecht Dürer's Rhinoceros in Zooological Literature," *Science, Medicine, and History: Essays on the Evolution of Scientific Thought and Medical Practice*, ed. E. Ashworth Underwood, 337-356. From page 71 of Edward Tufte's *Visual Explanations*.

Full-width text blocks

Many thanks go to Edward Tufte for leading the way with his work. It is only through his kind and careful editing that this project accomplishes what it does. All errors of implementation are of course mine.

Use pp to preprocess Markdown

We can use a list of macros defined below to generate a HTML or LaTeXfile depend on the format.

```
!comment(
title:  Tufte Markdown
author: David Peng
date:   2017-06-30
)

!ifeq(!format)(html)
(
```

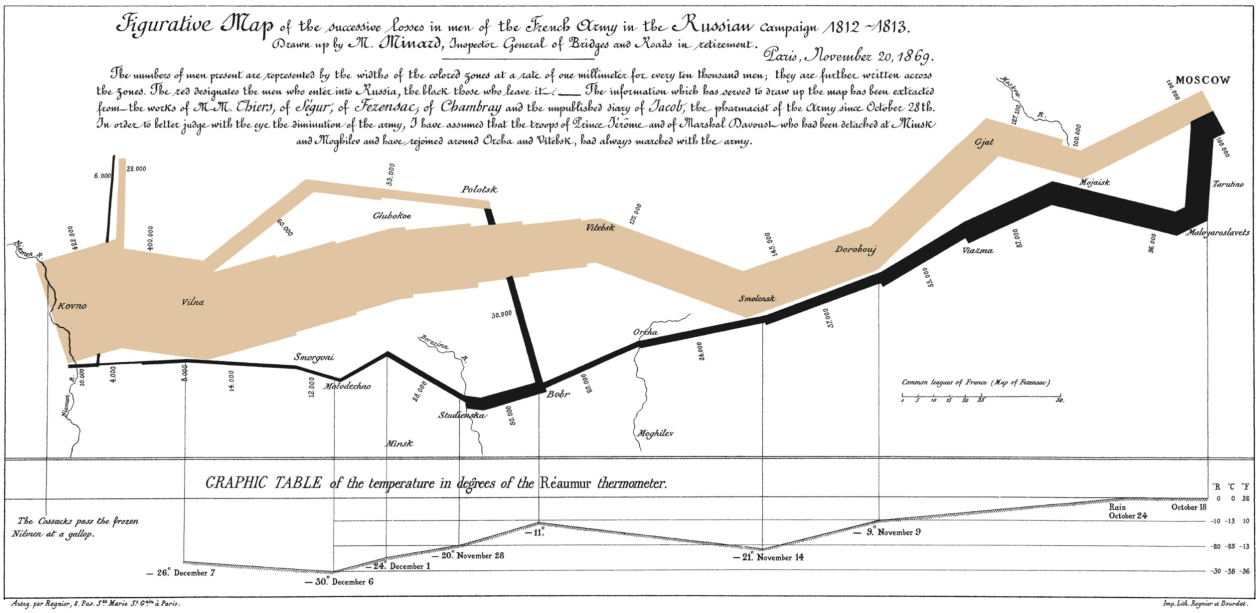


Figure 1: Figurative map of the successive losses of the French Army in the Russian campaign, 1812-1813

```
!import(html/tufte/newthought.html.pp)
!import(html/tufte/sidenote.html.pp)
!import(html/tufte/marginnote.html.pp)
!import(html/tufte/figure.html.pp)
!import(html/tufte/marginfigure.html.pp)
!import(html/tufte/fullwidthfigure.html.pp)
!import(html/tufte/fullwidth.html.pp)
)

!ifeq(!format)(pdf)
(
!import(pdf/tufte/newthought.pdf.pp)
!import(pdf/tufte/sidenote.pdf.pp)
!import(pdf/tufte/marginnote.pdf.pp)
!import(pdf/tufte/figure.pdf.pp)
!import(pdf/tufte/marginfigure.pdf.pp)
!import(pdf/tufte/fullwidthfigure.pdf.pp)
!import(pdf/tufte/fullwidth.pdf.pp)
)
```

Use Pandoc to convert Markdown to HTML and PDF

```
#!/bin/bash
```

```
# Build README
pp -html -import=./pp-macros/html/common/today.html.pp README.pp.md >README.md

# Build sample handout PDF
pp -pdf -import=./pp-macros/all.pp sample-handout.md | \
pandoc -o sample-handout.pdf -f markdown+raw_tex --pdf-engine=xelatex \
--template=./templates/tufte-handout.tex -V documentclass=tufte-handout \
--no-highlight

# Build sample handout HTML
pp -html -import=./pp-macros/all.pp sample-handout.md | \
pandoc -o sample-handout.html -s --template=./templates/tufte.html5 \
--no-highlight

# Build common PDF
pp -pdf -import=./pp-macros/common.pp ./test/common.md | \
pandoc -o ./test/common.pdf -f markdown+raw_tex --template=default.latex \
--pdf-engine=xelatex -V graphics=true \
-V strikeout=true -V "header-includes=\usepackage{soul}" \
-V "header-includes=\usepackage{color}"

# Build common HTML
pp -html -import=./pp-macros/common.pp ./test/common.md | \
pandoc -o ./test/common.html -t html5 -s \
-c https://cdnjs.cloudflare.com/ajax/libs/marx/3.0.3/marx.min.css \
-V "include-before=<main>" -V "include-after=<//main>"
```

Acknowledgements

- Tufte CSS
- RStudio Tufte Handout
- R Markdown Tufte Style
- RStudio Pandoc template: tufte-handout.tex