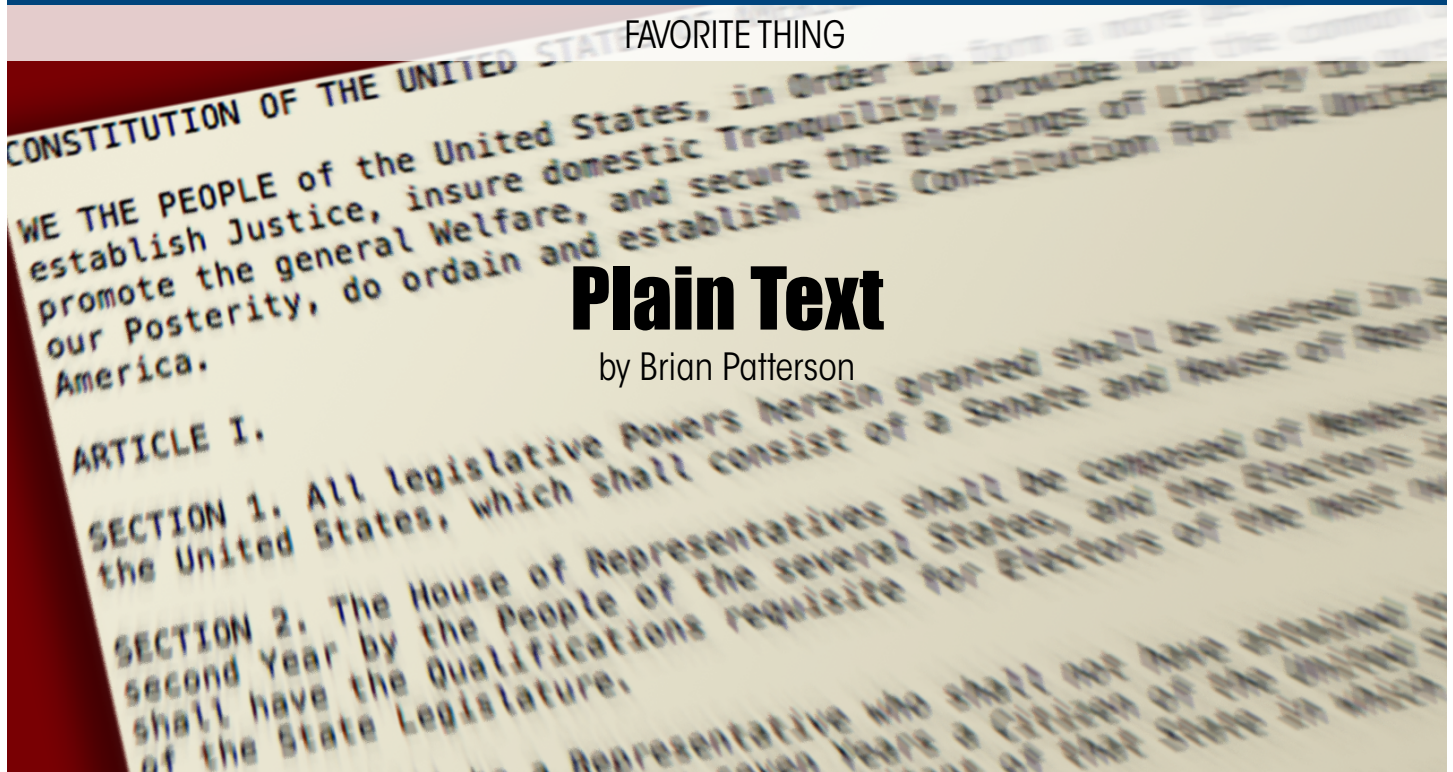




FAVORITE THING



# Plain Text

by Brian Patterson

**P**LAIN TEXT IS THE COCKROACH of file types: it will outlive us all. If you've ever tried opening a document from an old or obsolete word processor or page layout program, you know how difficult, if not impossible, it is to do. If you can find software that will actually open it, you might be lucky enough to see something you recognize hidden among the computer gibberish that surrounds it. Digging through the rubble to salvage what you can may eventually get you something useful, but it's no one's idea of a good time.

On the other hand, if you have any old .txt files you want to browse, they'll open easily and look exactly the same as they did before. There will be no unrecognizable characters added, no weird computer noise, just simple, plain text. Plain text will always survive because it is free from computer formatting, more or less, and is instead human-readable.

Best of all, plain text editors are everywhere, and many of them are free. TextEdit comes preinstalled on Macs, as does WordPad on the PC. [TextWrangler](#) is free and, in addition to text editing, offers very powerful tools for manipulating text.

You can write in plain text without worrying about what it will eventually look like. It is sort of like writing in long hand, except with a keyboard and screen instead of a pen and paper. The content is the center of attention, not which font to use, how big the font should be, how far apart the lines should be, or the millions of other choices a program like Microsoft Word puts in front of us.

Many things you write never have to go beyond plain, unformatted text. Notes (maybe organized with a program such as [Simplenote](#) or [Drafts](#)) or todo lists may never leave their humble beginnings. Some, however, will require stylized text, tables, footnotes, or images, which plain text can't deliver on its own. And if you intend on publishing on the web, in an ebook, in print, or maybe all three, you will need to provide direction on how you intend for your document to be formatted. The instructions on how your document is structured are called markup.

The slashes and letters in the above text are instructions on how the text should be styled, though the styles are not defined within the document. This could be considered human-readable

markup, because although there are characters that wouldn't normally be where they are, the text is generally recognizable.

While it has the advantage of a front end editor keeping the code hidden away underneath, RTF has very minimal formatting options available. There are other, better options available if you are willing to spend a few minutes getting your hands dirty.

**Markdown** is a very minimal [markup language](#) designed for writing for the web. It is designed to be compatible with HTML, yet as readable as possible to non-coders. To quote its creator John Gruber:

“The idea for Markdown is to make it easy to read, write, and edit prose. HTML is a publishing format; Markdown is a writing format. Thus, Markdown's formatting syntax only addresses issues that can be conveyed in plain text.”

It is easy to [learn](#) and because it is written using plain text, it is much simpler yet more precise than a [WYSIWYG](#) editor. [MultiMarkdown](#) is another version which encompasses and extends Markdown with more options. And the options for text editors geared specifically toward Markdown and other mark-

up languages is [ridiculous](#). Here are just two of the many other markup languages worth exploring. [Textile](#) was introduced in 2003 and offers more formatting options than Markdown, but still with simple syntax. This article is being written in Textile. [LaTeX](#) is a markup language with a somewhat steeper learning curve, but it has powerful controls, many geared toward the scientific and academic communities. It produces beautifully typeset PDFs for anyone willing and able to implement it.

What makes plain text my favorite, though, is the ability to translate it between these markup languages automatically using a converter like [pandoc](#). For this issue of [The Jury Expert](#), I used pandoc to batch convert all of the authors' articles and replies from Microsoft Word into Textile. Then I used [TextMate](#) as my editor to format the documents. TextMate has its own Textile to HTML converter that I used to create the HTML which went onto [The Jury Expert](#), though I could have used pandoc for that as well. Finally, I exported the HTML to a variation of XML that is readable by Adobe InDesign called ICML.

That's a long way around to say that plain text is a great tool, more powerful than many realize, and it's worth considering how it might be better suited to some of the things you do than what you are currently using. ©

***Brian Patterson is a graphic designer and trial consultant at [Barnes & Roberts](#). He has created and overseen production of multimedia presentations for well over a hundred courtroom proceedings since 1998.***