

# Markdown: The Syntax You (Probably) Already Know

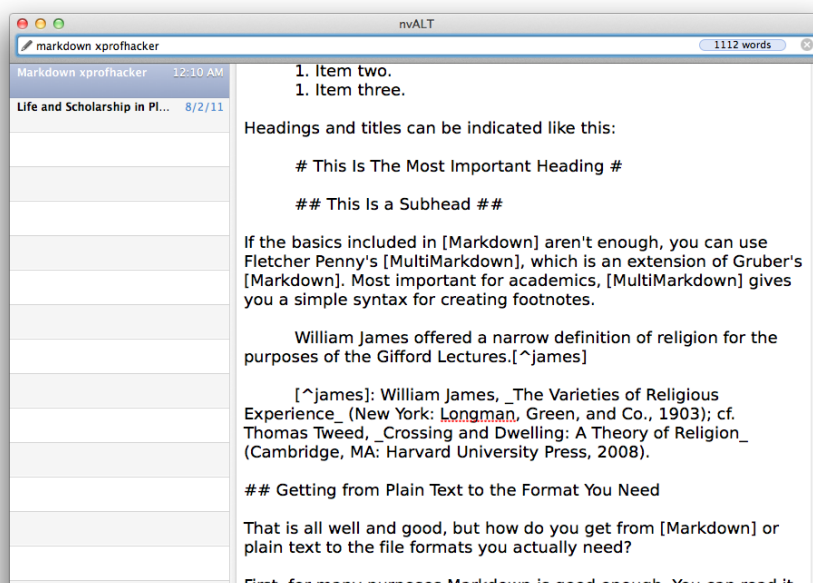


Figure 1: Writing in Markdown using nvALT

There's a simple idea about productivity and writing: almost anything you need to do, you can do in a plain text file. By a plain text file, I mean a .txt file that contains nothing but text without formatting—the kind of file that your PC probably opens in Notepad and that your Mac opens in TextEdit. (Linux users already know what I'm talking about.)

There are a lot of reasons to love plain text. We've mentioned some of them in earlier ProffHacker posts, such as Alex's about JDarkRoom and Jason's about nvALT. Briefly, plain text is a great format to use because (1) it can be read by any computer or device; (2) it's future proof, since computers will always be able to read it; (3) it can be synced to all your devices; (4) it can be converted to virtually any format. In a future post, I plan to explain the virtues of plain text at greater length.

But there is one major objection to writing in plain text. In fact, it's an objection that ProffHacker commenter iredale has made:

The program saves only to .txt files? That's not a "word processor," it's a "text editor." And while it might work for techno-geeks who

write code, it's utterly useless for many of us in academia. With the exception of email, virtually everything I write requires formatting. And not just the basics — font, type size and spacing — but also things like footnotes, headers, and footers.

There's a way for plain text to handle formatting too, in a way that's lightweight and so easy to learn, you might already know it.

## The Syntax You Already Know

Suppose you're writing an e-mail (that will be sent in plain text). You want to mention a book to a student, but because it's plain text you can't *italicize* the title. What do you do? You would probably use underscores, like this:

```
Be sure to read William James, _The Varieties of Religious Experience_.
```

That looks like a title. But you also want to emphasize a point, using **bold** if it weren't plain text. So you might write something like this:

```
James's book will **definitely** be covered on the final exam.
```

That gets the point across. And of course, you'll want to quote something from the book, so you might invent a way to format a block quote, borrowing from how your e-mail program formats quotes:

```
You must be able to discuss James's definition of religion:  
> Religion, therefore, as I now ask you arbitrarily  
> to take it, shall mean for us the feelings, acts, and  
> experiences of individual men in their solitude,  
> so far as they apprehend themselves to stand in  
> relation to whatever they may consider the divine.
```

And you'll also want to include a link to the book, and it would be nice to distinguish the link from the text, so you use angle brackets ( < and > ):

```
James's book is available in full text at Google Books <http://books.google.com/books?ic
```

We've invented a simple syntax that handles basic typography even in plain text. If that syntax makes even a little bit of sense to you, the congratulations: you already understand Markdown.

## Markdown

Markdown is a syntax invented by John Gruber for marking up plain text. It's a markup language, like HTML, in that you indicate the structure and meaning of parts of a document. It's unlike HTML, in that Markdown is meant to be read by humans, rather than machines.

So Markdown's syntax is very simple. In addition to the examples above, you can make lists using bullets or numbers:

```
* An item.
* Another item.
* Yet another item.

1. Item one.
1. Item two.
1. Item three.
```

Headings and titles can be indicated like this:

```
# This Is The Most Important Heading #

## This Is a Subhead ##
```

If the basics included in Markdown aren't enough, you can use Fletcher Penny's MultiMarkdown, which is an extension of Gruber's Markdown. Most important for academics, MultiMarkdown gives you a simple syntax for creating footnotes.

```
William James offered a narrow definition of religion for the purposes of the Gifford Lectures.
```

```
[^james]: William James, _The Varieties of Religious Experience_ (New York: Longman, 1902).
```

## Getting from Plain Text to the Format You Need

That is all well and good, but how do you get from Markdown or plain text to the file formats you actually need?

First, for many purposes Markdown is good enough. You can read it, edit it, and e-mail it to a colleague as easily as you can any other file format. And at the very least, you can copy and paste plain-text anywhere (but good luck getting your data out of other programs)

For many purposes, though, you'll need to move your text into a more complicated format. That's why Markdown is also the name of a Perl script that Gruber wrote to transform Markdown syntax into HTML to be put on the web.

If you know what to do with a Perl script, good for you. But if you're already on the command line, you should use Pandoc instead, which can convert between an amazing array of file formats, including Markdown, LaTeX, RTF, ODT, and HTML. (We'll cover Pandoc more in a later post.)

Or, if you're not comfortable with the command line, you can use any of a number of free or cheap programs. nvALT, which is the program I'm using to write this post, will let you export a document from plain text to HTML or PDF.

There is also a tutorial for another way of getting from Markdown to Word without using Pandoc.

At the risk of being too meta, I've put my original plain text file for this blog post, written in Markdown, along with the blog post converted to HTML, LaTeX, RTF, and ODT to a Github repository. I've also taken the examples I used and put them into separate files. You can browse those files to see how Markdown syntax ends up becoming other files.

**What about you? Are you living the plain text life? Have you used Markdown?**