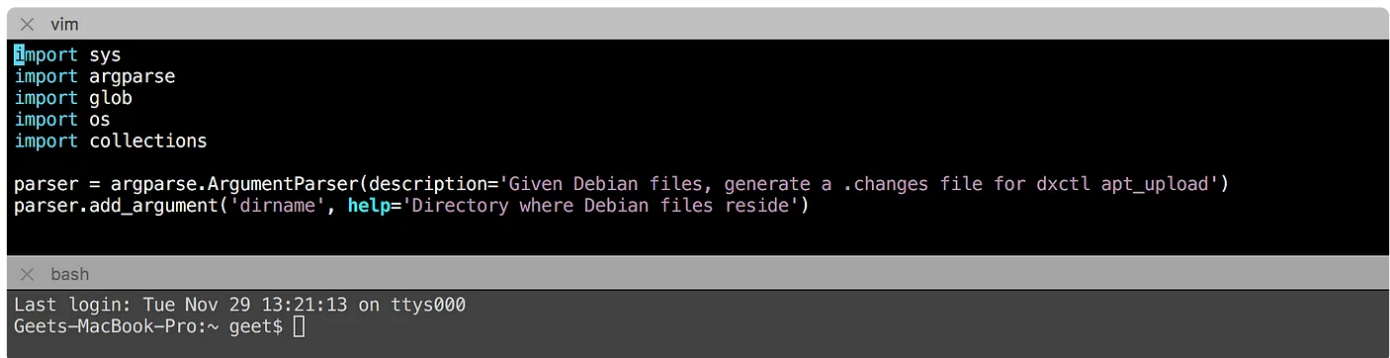


'Editor-First' vs. 'Terminal-First' for Development

[Geet Duggal](#) • 2 mins

[Dec 1st 2016](#)

The case for in-terminal editing with Vi/Emacs



```
vim
import sys
import argparse
import glob
import os
import collections

parser = argparse.ArgumentParser(description='Given Debian files, generate a .changes file for dxtl apt_upload')
parser.add_argument('dirname', help='Directory where Debian files reside')

bash
Last login: Tue Nov 29 13:21:13 on ttys000
Geets-MacBook-Pro:~ geet$
```

Great editors like Atom, Sublime, and GUI front-ends to Emacs/Vi are commonly used by developers.

They scroll smoothly, have all sorts of colorful features, and let you install sophisticated packages for development in multiple languages and version control systems.

Why would anyone go back to the stone age and edit within a terminal using Vi/Emacs?

I have had the same question many times before.

I have also tried to answer it by using editors like the ones I mentioned above.

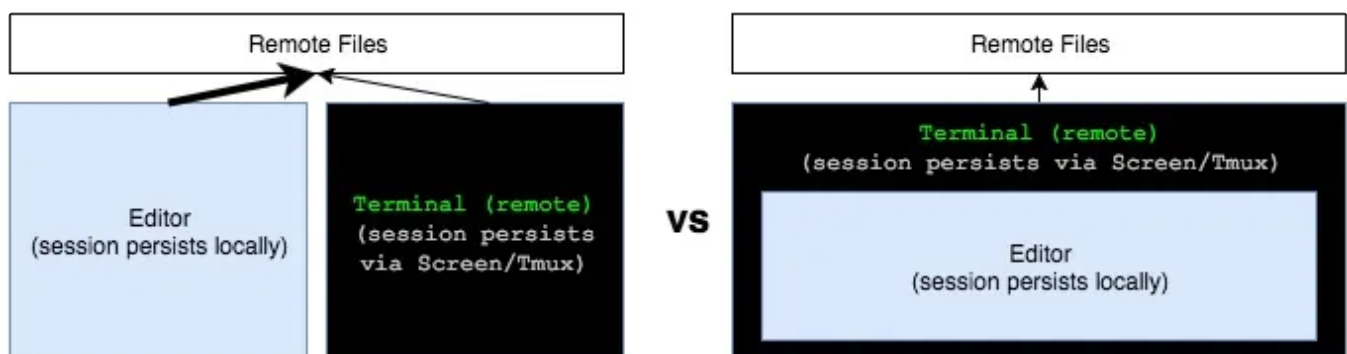
Inevitably, I end up wanting to develop on a remote server (e.g. I have a development AWS instance with some fancy features I am experimenting with).

So I end up spending an amusing amount of time setting up some remote connection facility in <insert your favorite editor here> (e.g. Tramp for Emacs, ...) in order to edit files remotely with my beautiful editor.

And inevitably, I always run into some issue where I need to reconnect or I get out of sync in a way that slows my flow down.

Also, development isn't just about code. I spend a substantial fraction of my time on the terminal, trying commands, testing, and exploring.

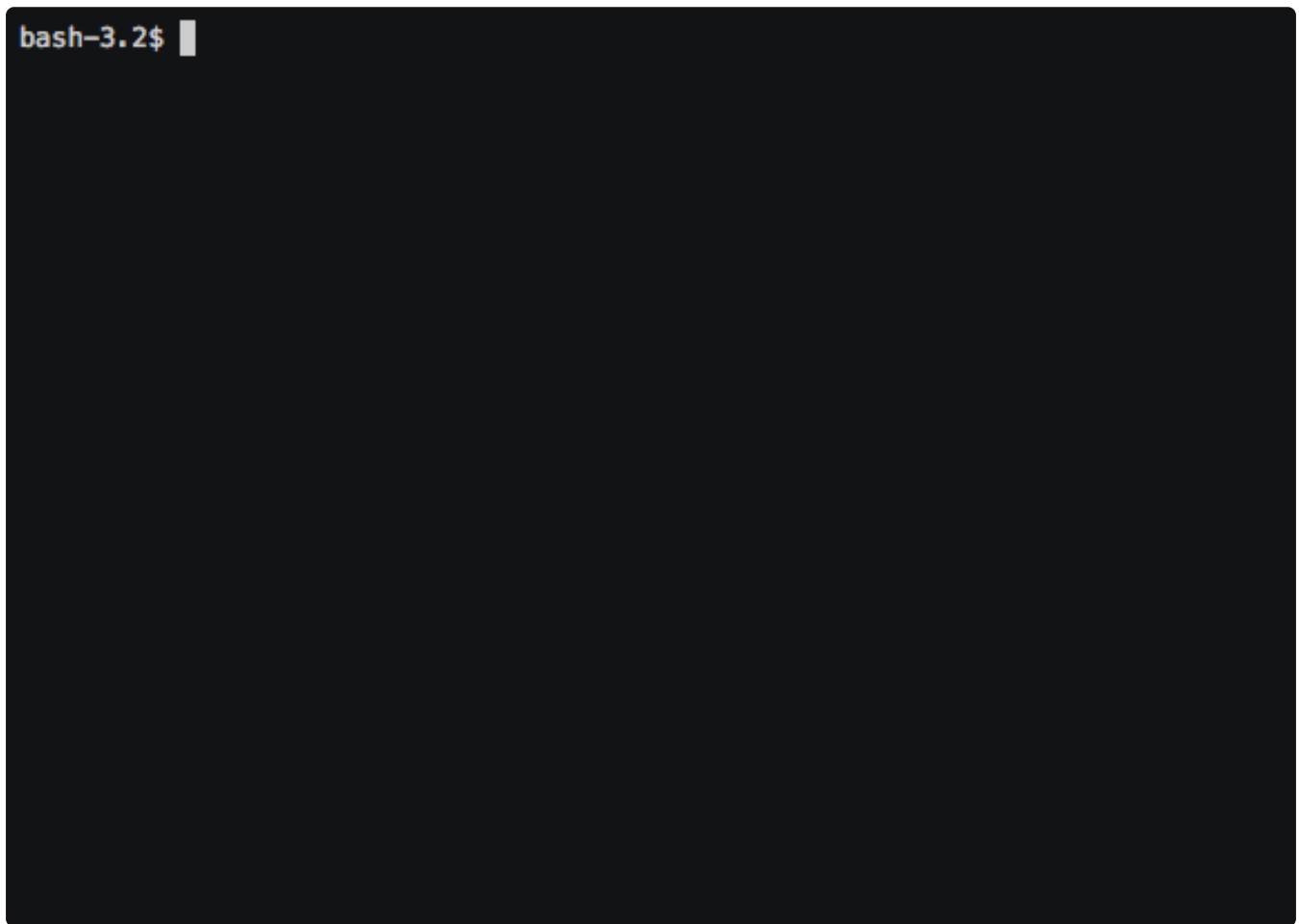
So I always end up abandoning this 'editor-first' approach for one that is 'terminal-first':



'Editor-first' vs. 'terminal-first' approach to development

(You can imagine the 'editor-first' approach displayed above getting a lot more complicated in case you end up going through another host before having to edit.)

Here is an example of my 'Vi flow' for 'in-terminal' development:



Simple example of using Vi in a terminal while going back and forth to a terminal

This system also has some additional perks. If I combine it with a tmux

session on a development node, I can easily hop back into the state of development where I left off, and because there's no fancy editing software, I can literally continue work anywhere with a terminal. I have successfully done this on my iPad pro, a Chromebook, and even my phone when I wanted to make a quick edit (Serverauditor is my terminal app of choice for iOS).

I am curious to what others do and especially curious if someone can convince me to stop using my stones to write on cave walls. Do you develop editor-first or terminal-first? Why?