

Vinn

Technical Workshop

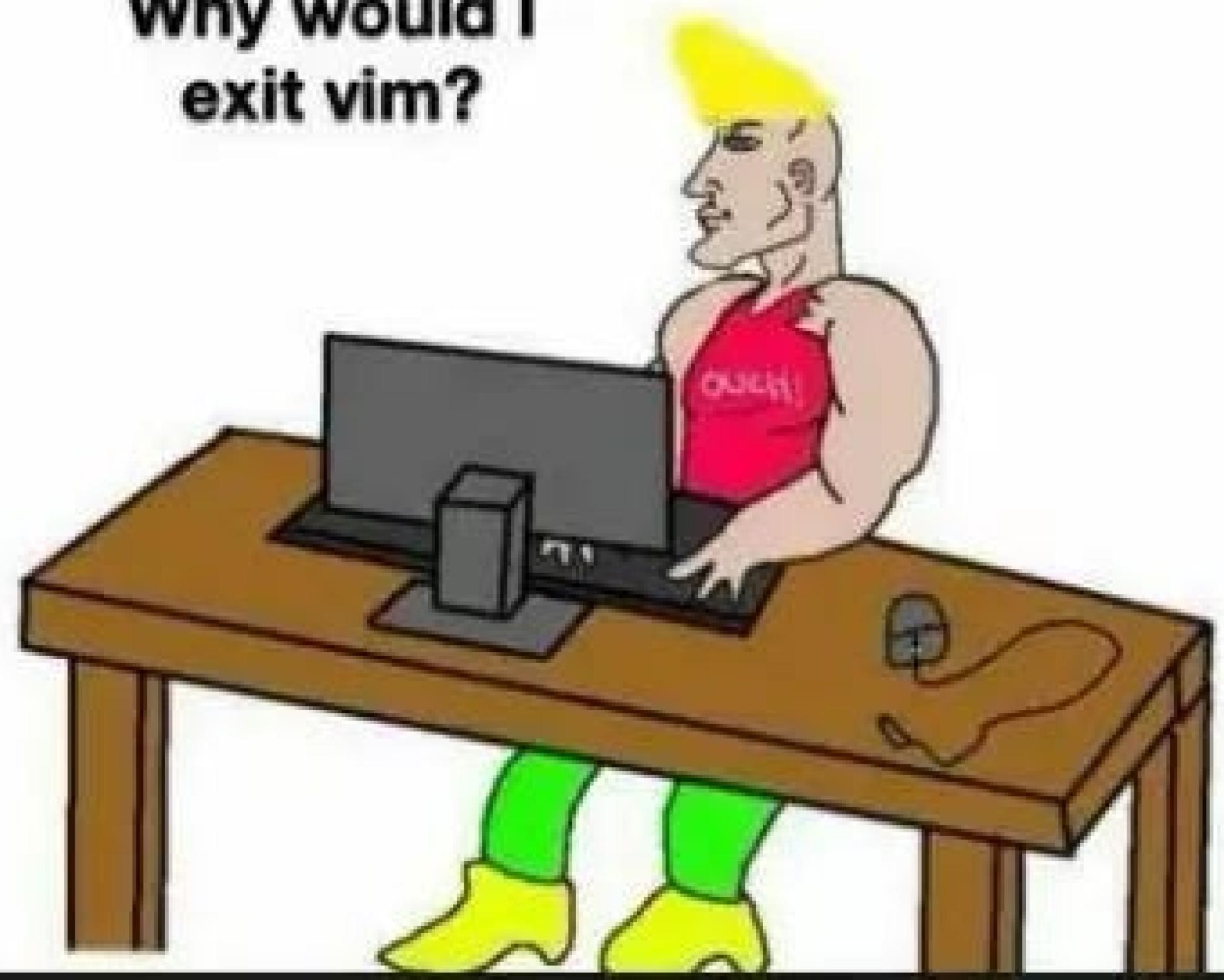
Will
we
ever
be
able
to
exit
Vim?



**How do I
exit vim?**

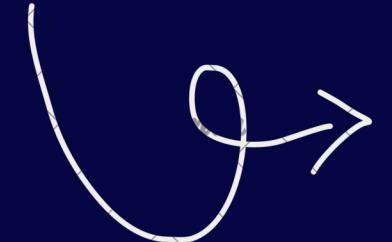


**Why would I
exit vim?**



What, Where, How

Vim (/vɪm/; vi improved) is a free and open-source, screen-based text editor program. It is an improved clone of Bill Joy's vi.



Vim is designed for use both from a command-line interface and as a standalone application in a graphical user interface.



Vim's interface is not based on menus or icons but on commands given in a text user interface; its GUI mode, gVim, adds menus and toolbars for commonly used commands but the full functionality is still expressed through its command line mode.

Why

When programming, you spend most of your time reading/editing, not writing. For this reason, Vim is a modal editor: it has different modes for inserting text vs manipulating text.

Vim avoids the use of the mouse, because it's too slow; Vim even avoids using the arrow keys because it requires too much movement.

The end result is an editor that can match the speed at which you think.



Agenda:

- Set-up Vim
 - Workshop : <https://github.com/acm-uic>
- Modal Editing
- Movements
- Edits
- Buffers



Modal Editing:

Normal: for moving around a file and making edits

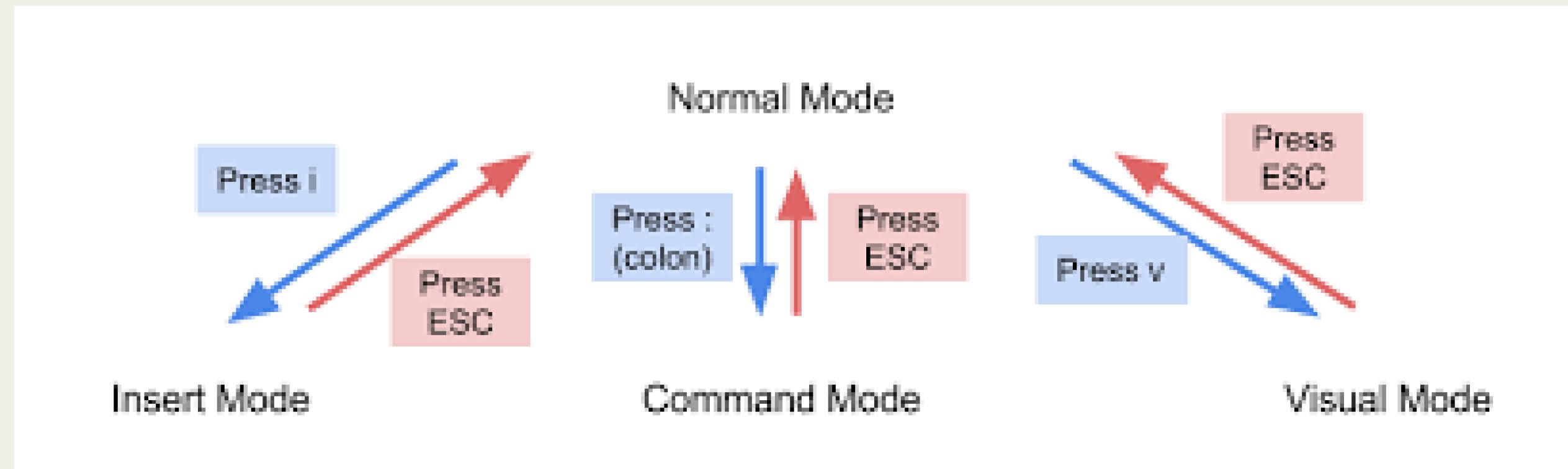
Insert: for inserting text

Replace: for replacing text

Visual (plain, line, or block): for selecting blocks of text

Command-line: for running a command

4



Deep-diving into modes:

Normal (default):

Used for:

- Moving the cursor
- Deleting characters, words, lines, etc
- Performing operations like Copy/Paste

Insert (press ‘i’):

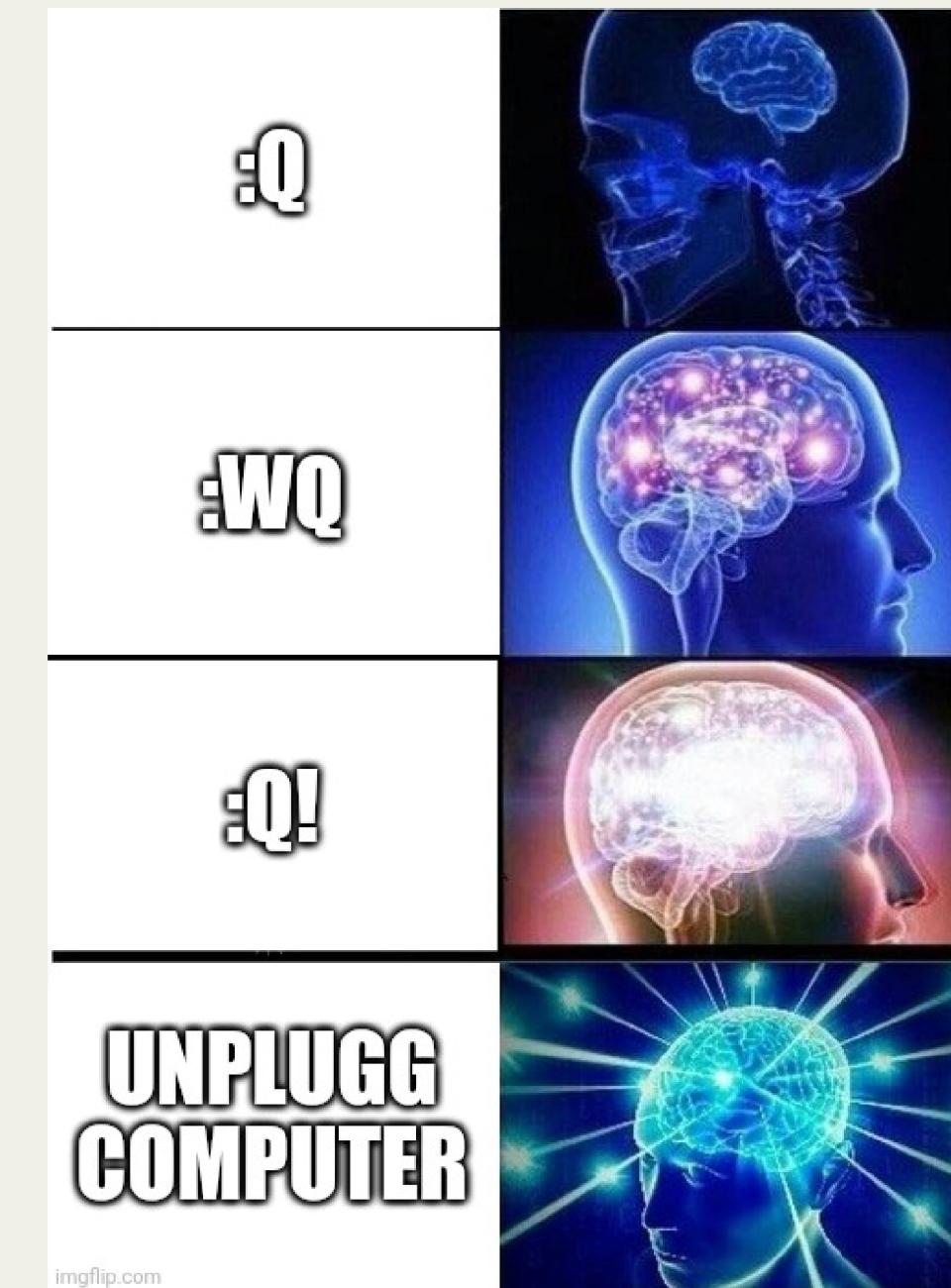
- Used for editing and inserting text.
- You can think of insert mode as the default behavior in the editors you are already used to.

Deep-diving into modes:

Command mode (press ':'):

When you enter command mode, your cursor will jump to the command line at the bottom of the screen. This mode has many functionalities, including opening, saving, and closing files, and **quitting Vim**.

- :q quit (close window)
 - will prompt for unsaved changes!
- :w save (“write”)
- :wq save and quit
 - recommended
- :e {name of file} open file for editing
- :help {topic} open help



Deep-diving into modes:

Visual (press 'v'):

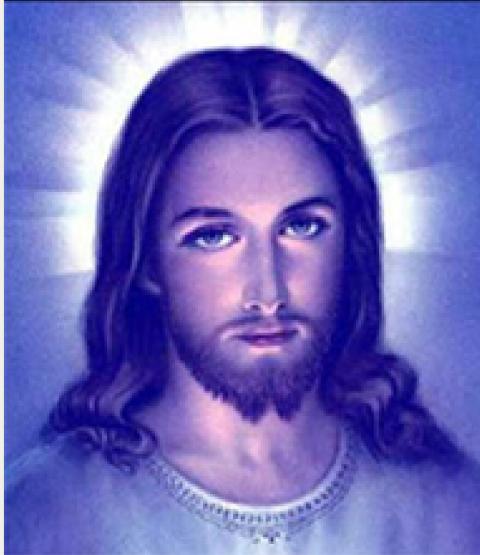
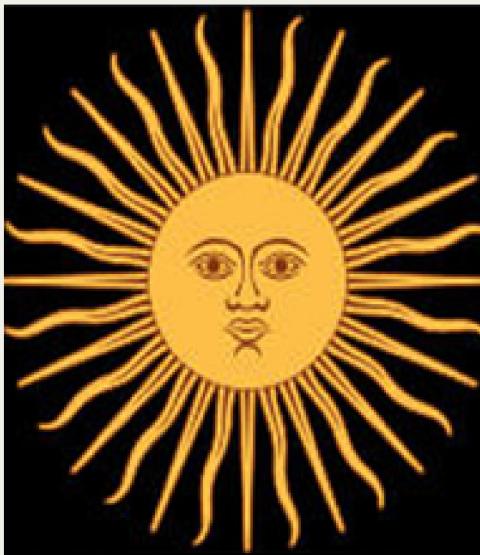
Visual modes:

- Visual: v
- Visual Line: V
- Visual Block: Ctrl-v
- Can use movement keys to make selection.





Google



Movements:

Basic movement: hjkl (left, down, up, right)

Words: w (next word), b (beginning of word), e (end of word)

Lines: 0 (beginning of line), ^ (first non-blank character), \$ (end of line) 9

Screen: H (top of screen), M (middle of screen), L (bottom of screen)

Scroll: Ctrl-u (up), Ctrl-d (down)

File: gg (beginning of file), G (end of file)

Line numbers: :{number}<CR> or {number}G (line {number})

Misc: % (corresponding item)

Find: f{character}, t{character}, F{character}, T{character}

Search: /{regex}, n / N for navigating matches



Edits:

- o / O insert line below / above
 - d{motion} delete {motion}
 - e.g. dw is delete word, d\$ is delete to end of line, d0 is delete to beginning of line
 - c{motion} change {motion}
 - e.g. cw is change word
 - like d{motion} followed by i
 - x delete character (equal do dl)
 - s substitute character (equal to cl)
 - Visual mode + manipulation
 - select text, d to delete it or c to change it
 - u to undo, <C-r> to redo
 - y to copy
 - p to paste
- NOTATION: All of below mean Ctrl + V
 - ^V
 - Ctrl-V
 - <C-V>

Buffers:

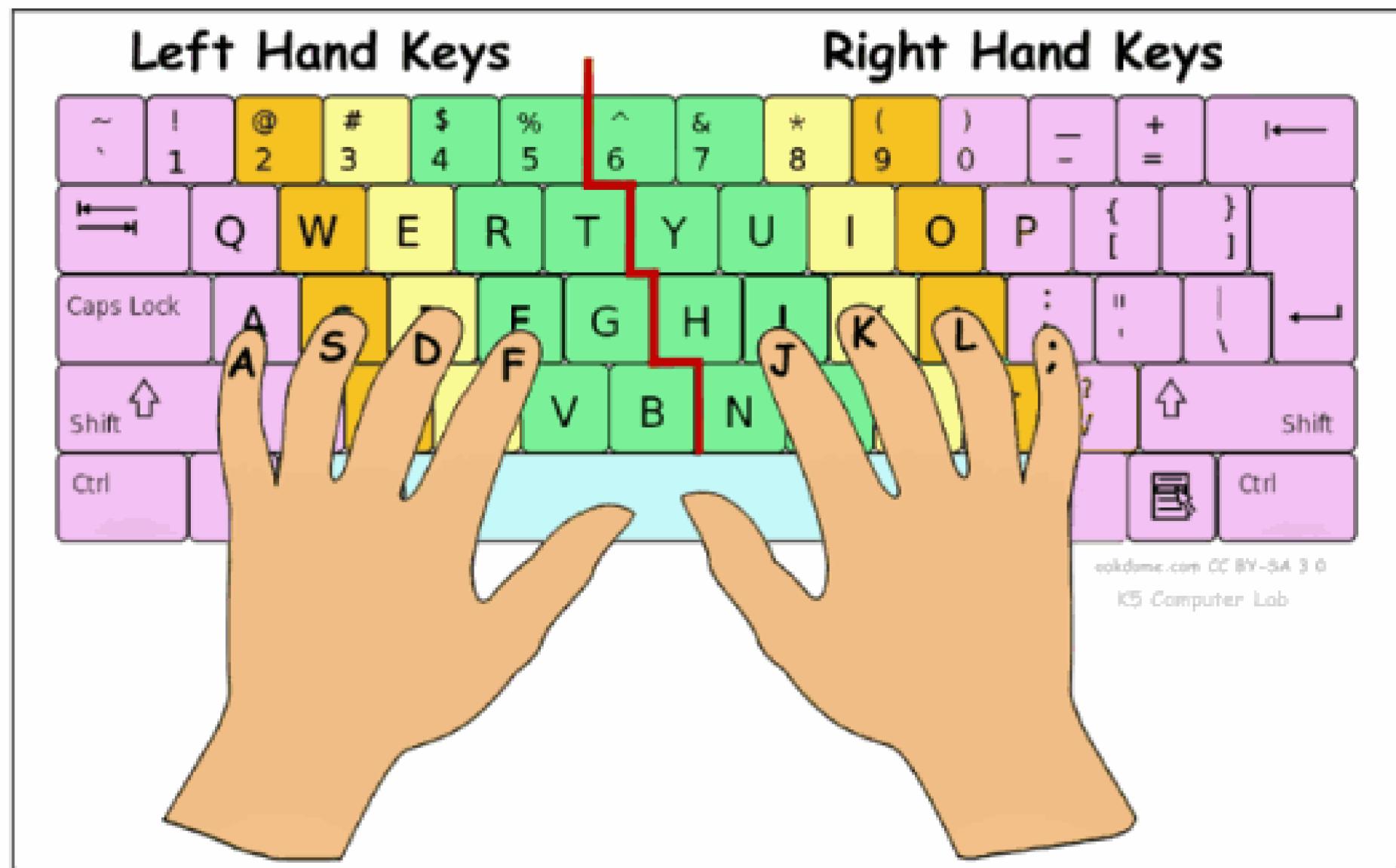
Vim maintains a set of open files, called “buffers”. A Vim session has a number of tabs, each of which has a number of windows (split panes).

Each window shows a single buffer. Unlike other programs you are familiar with, like web browsers, there is not a 1-to-1 correspondence between buffers and windows; windows are merely views. A given buffer may be open in multiple windows, even within the same tab. This can be quite handy, for example, to view two different parts of a file at the same time.

By default, Vim opens with a single tab, which contains a single window.

Sport mode:

Proper Finger Placement on the Keyboard



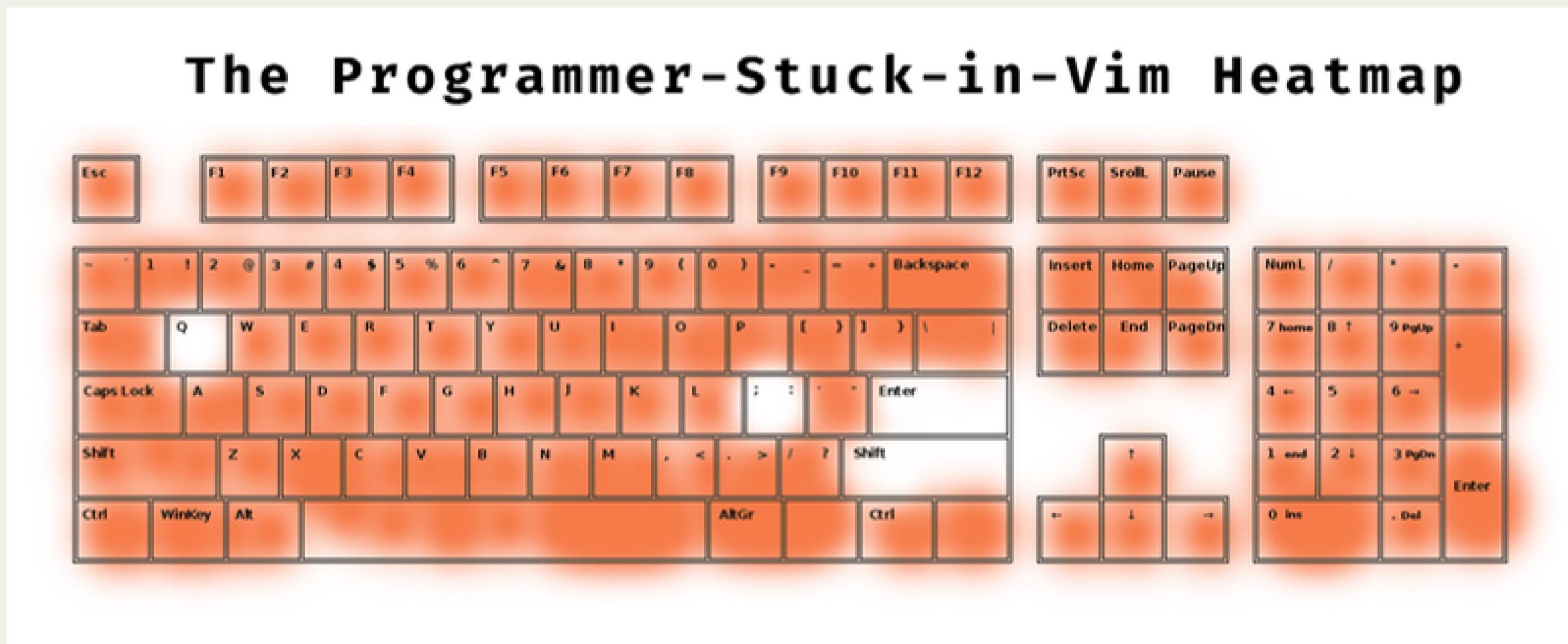
Flex:

Create **program.c** with your name and a fun fact about yourself!

```
#include <stdio.h>

int main(int argc, char **argv) {
    printf("Welcome to ACM!\n");
    printf("I am Yashi Shandilya\n");
    printf("I am the President, Tech Lead, and SIG Lead for SIG: Jobs and AI\n");
```

Exit with **ESC** key + **:wq** / **:q!**



Relevance:

Philosophy

Why did we even
bother with all of this?

Basics

Modal editing with all
different kinds of
modes

Advanced

Movements
Edits
Buffers
Sport mode!
Ready for CS 261 & CS 361

Resources for the curious:

Games:

<https://vimsnake.com/>

<https://vim-adventures.com/>

Interactive:

<https://vim.is/#exercise>

<https://www.openvim.com/sandbox.html>

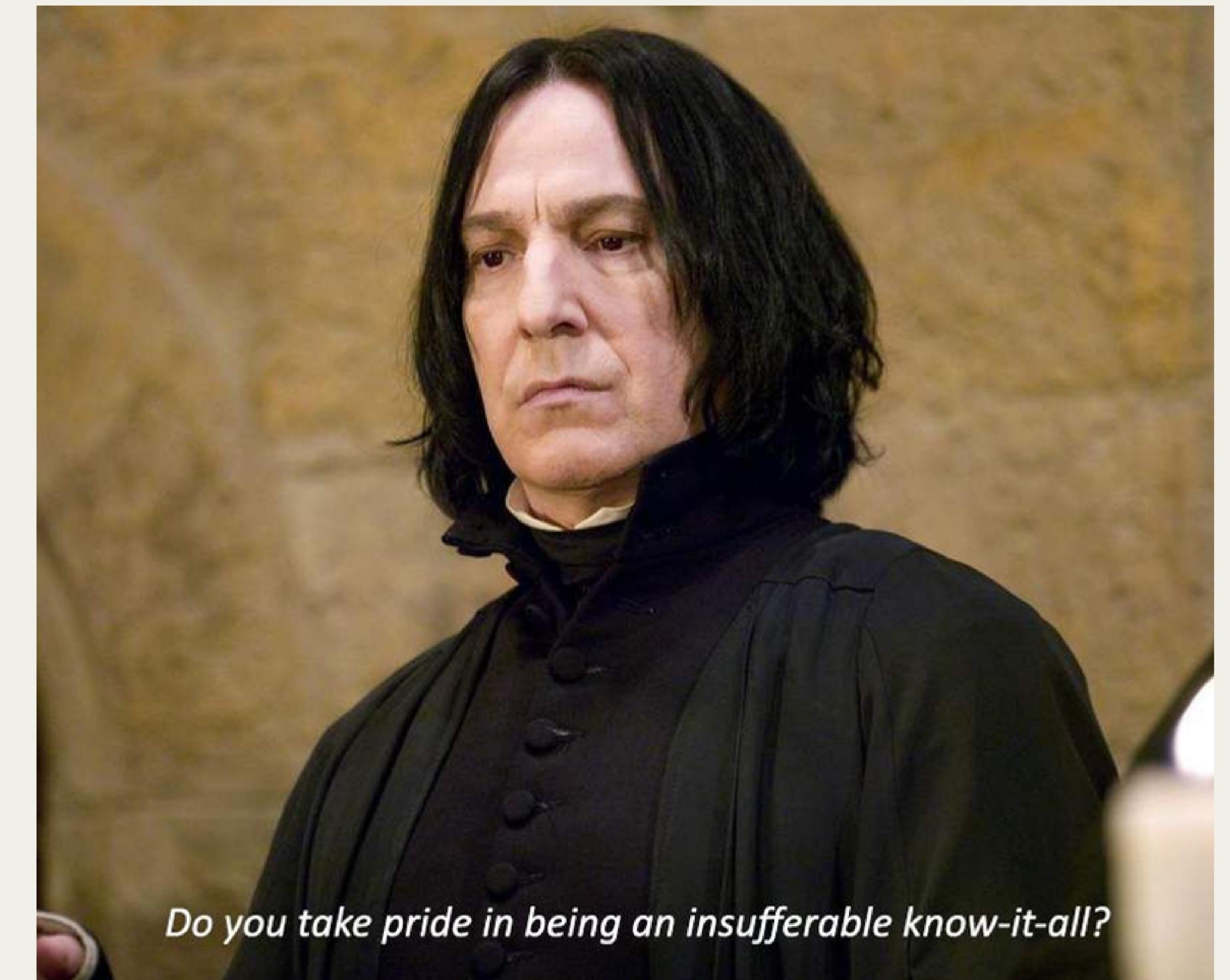
<https://www.vim-hero.com/>

Lectures:

<https://missing.csail.mit.edu/2020/editors/>

Cheatsheet:

<https://vim.rtorr.com/>



Do you take pride in being an insufferable know-it-all?

You're Welcome!

Ping me
@mentossuperfresh
for any questions!



Join ACM!

14