The standard simple text editors (notepad, gedit, nano, TextEdit, etc.) are good enough to get by, but clunky when you really want to edit code.

Most programmers, and most scientists, use vi and/or emacs. These are unix text editors widely considered to be "more powerful" and all around better.

In this context, "powerful" means there are some things you can do in these text editors that you simply could not in others.

We're going to use the advanced version of vim called `gvim`, or graphical vim. `vim` stands for vi *improved*, and `vi` is short for visual. Unix gurus are all about minimizing the number of keystrokes required to get things done.

Be sure to read the instructions before entering commands! Some of the commands are shown, but not meant to be run!

To start gvim, run it in the background:
`gvim &`

You'll be faced with a wall of white, with a bunch of `~`'s down the left side and a little information in the center.

Right now, if you try to enter any text, you're unlikely to see it, and weird things will start happening. You're in "command mode". `vim` has two main modes: command mode and insert mode. In command mode, you can only enter commands, while in insert mode, `vim` behaves like a normal editor.

If you tried entering any text, press `esc` now to get back to command mode. `esc` always returns you to command mode.

Now enter insert mode by pressing `i`. You should see the text `-- INSERT --` in the bottom left corner of the screen.

Type `Inserted some text.` then press `esc` to return to command mode.

At this point, we'll do some commands. First, and most important, is saving. Saving is referred to as "writing to disk", so the command to save is `:w`. Type:
`:w vim_tutorial.txt`
You should see those letters show up in the bottom-left corner. Then press enter. You have to press enter after colon commands (things that start with `:`), but you don't press enter after any of the others unless it says `<enter>` afterwards.

Colorscheme: Change the colorscheme. Type `:colorscheme elflord <enter>`. If you really dislike these colors, `:colorscheme default <enter>` will reset them.

Copy & Paste:
To copy text, you use the yank command. To copy a whole line, type `yy`. Do this now. Then, type `p` to paste. You should now have two identical copies of the first line.

What about selecting text? You can do this with the mouse or the keyboard; we'll try both. Select the whole of the second line by clicking & dragging with the mouse. You should see `-- VISUAL --` in the bottom left corner of the screen now. With the text highlighted, you can press `y` just once and it will copy the selected text. Do this.

Now, type `P` - this will paste *before* the cursor. Your file should now look something like this:

```
Inserted some text.
Inserted some text.Inserted some text.
```

Make sure your cursor is at the start of line 2. Now, we'll copy the word "Inserted": type `yw` (short for "yank word").

We'll paste this on the next line. To make a new line, type `o` (lower-case O). This will drop you into insert mode, so press `esc` to get back to command mode.

Now, let's paste the word 80 times. Type `80p` (just like that - 8,0,p).

Deleting:
That last line is pretty silly. To delete a word, you can use `dw`. Or we could delete 80 words with `80dw`. Don't do this, though! Instead, let's delete the whole line:
`dd`

Line Numbers:
To show line numbers, in command mode, enter:
`set number`

This is really nice for editing code.

Case Changing:
Move your cursor to the start of line 1. Press `v` to enter visual mode. Now, move your cursor to the right (with the directional pad) until the word "Inserted" is highlighted. We'll change the case of all the letters in the word: press `~`.

Visual Block Mode:
To get to the start of line one, do the following (no commas): `gg` , `^`
Explanation: `gg` moves to the top of the document, and `^` moves to the beginning of the line. (`$` moves to the end of the line)

Now, move the cursor ahead one word by typing `w`. `w` moves ahead one word, `b` moves back one word, and `e` moves to the end of a word.

Press `control-v` to enter visual block mode, then `e` to go to the end of the word "some", then press the down-arrow to move down a line. You should have selected the word "some" twice. Now press `y` to copy them.

Visual blocks are really cool and unique to vim.

Press `P` to paste the block.

We're not done with visual block yet. Type the following commands (no commas):
`gg` , `2yy` , `P` , `p`

You should now have 6 lines in your file. We're going to "comment them out".

Type `gg` then `^` to move to the beginning of the file. Now press `control-v` to enter visual block mode again, and then `G` to move to the *end* of the file.

Now, press `I` (capital I) to enter insert mode. Type `;` to comment out the line. Then press `esc`. Now, ALL of your lines should be preceded by a `;`.

Searching:
VIM has a really nice search feature. Type `/` to enter search mode, then `sert <enter>` to search for the string `sert`. You should see all instances of `sert` (in the word insert) highlighted, but the capitalized `SERT` will not be.

Search and Replace:
This is the real power of vim, but it's kinda complicated. We'll start easy: replace all instances of `sert` with `ject`. This is the command to do that:
`:%s/sert/ject/g`
`%` means all lines, `sert` is the text to replace, `ject` is what to replace it with, and `g` means replace ALL instances on a given line.

Quitting:
To quit, you type `:q <enter>`. To save and quit, type `:qw <enter>`. Save and quit! Then, move `vim_tutorial.txt` into your tutorials directory on github and turn it in.