Editing files with vi

The thing you have to understand about **vi** and its work-alike editors is *modality*. Most programs have just one mode, accepting input and placing it at the cursor. This is what you probably expect from a program. But `vi` has other modes. When you start `vi`, you'll be in "Normal" mode, which is really *command* mode. When you are in Normal mode, whatever you type is considered *not* to be input, but *commands* that vi will try to execute.

This may sound a little crazy, but it is actually a very powerful way to edit documents. Even if you hate it, the ubiquity of `vi` means that you're going to need to learn the basics, because sometimes you just *have* to use it. On the other hand, if you enjoy working at a command line, then you may end up *loving* `vi`.

Lesson 1: How to Quit vi

Since `vi` is the default editor, there's a good chance you got dropped into it without knowing it. This can be *extremely* frustrating if you don't know your way around the program.

To exit `vi` *without* saving changes, hit `Escape` a couple of times to ensure you are in Normal mode, and then type:

```
:q!
```

Lesson 2: Editing with `vi`

Start `vi` and press `i`. The cursor may change shape, and `INSERT` is displayed at the bottom of the screen (in most `vi` clones). You are now in *insert mode* — all your keystrokes are entered into the current buffer and are displayed to the screen.

Hit the `Escape` key. The cursor changes shape again, and `INSERT` has disappeared. You are back in Normal mode. Hitting `Escape` a few times will pretty much always cancel whatever you are doing and return you to Normal mode.

Command mode is also where you move around in the file. On most systems, you can use the arrow keys to move the cursor. If that fails, cursor movement can be accomplished with the *hjkl* keys:

```
h   move left one character
j   move down one character
k   move up one character
l   move right one character
```

`vi` has its own command line, which you access by typing a colon. Commands typed at the command line will not be executed until you hit `Enter`, just like in the shell.

Many of the commands that you will use in command mode begin with a colon. For example, the command to quit is `:q`, as we learned in Lesson 1 above. Actually, in Lesson 1 we added `!`, which tells `vi` to "force" the operation. If you have edited the file, typing `:q` will *not* immediately exit the program, but instead

produce this error message:

```
E37: No write since last change (add ! to override)
```

To save your changes, use the `:w` command ("Write"). You can save and quit all in one go by typing both commands together:

```
:wq
```

Of course, if you started `vi` without giving a file name, you'll need to provide one. For example, to save your working file as `test.txt`, you would type:

```
:w test.txt
```

Try editing some text now.

> Start vi
> Hit `i` to go to Insert mode.
> Type some text.
> Hit `Escape` to return to Normal mode.
> Type `:w test.txt` to save your work
> Type `:q` to quit

`vi` Quick Reference

Movement

`h, j, k, l`
> left, down, up, right

`$`
> To the end of the line

`^`
> To the beginning of the line

`G`
> To the end of the file

`:1`
> To the beginning of the file

`:47`
> To line 47

Editing

`dd`
> Remove a line

`5dd`
> Removing five lines

`r`
> Replace a character

`x`
> Delete a character

`5x`
> Delete 5 characters

`u`

> Undo last action

`J`

> Join current and next lines (Note the capital — hold the `Shift` key)

Saving and Quitting

`:q`

> Quit

`:q!`

> Quit without saving

`:wq`

> Write and quit

`:w`

> Write (without quitting)

`:e!`

> Reload currently open file from disk
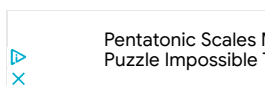
`:w test.txt`

> Write buffer to file `test.txt`

`:e test2.txt`

> Open file `test2.txt`

Additional Resources

vim documentation online
Learning the vi and Vim Editors
VI Editor Pocket Reference