

# vi (Vim) Text Editor

---

As you go further in your programming journey, you will realize how convenient it is to have a terminal text editor right there in the shell to edit documents and programs. You are familiar with various WYSIWYG word processing applications like Word and Google Sheets, which you can't use for writing code, and GUI text editors like Text Pad or Text Edit, which live outside of the terminal and require a mouse. When a programmer is developing in the console or terminal environment, it can become burdensome to keep going back and forth between the terminal and the desktop to make edits. That is why professional programmers use terminal text editors like Emacs or Nano. For this unit, we will be learning a standard UNIX editor called vi or sometimes referred to as Vim. Master the vi text editor and learn how to make complex edits on your files with less time and effort.

In this section we'll look at a tool to put content into files and edit that content as well. Vi is a text editor that is most likely very different to any editor you have used before. It will take a while to get your head around but once you do you will realize it is actually quite powerful. Even if you don't use Vi all the time you will definitely find that work patterns you develop in learning the editor can be transferred easily to other programs and to great effect. I've chosen to look at Vi first, not because it is the most popular, but I believe it is the easiest to learn. The aim is not to cover everything that Vi can do but to get you up and running with the basics.

Everything in Vi is done via the keyboard. There are two modes in Vi. Insert (or Input) mode and Edit or Command mode. In input mode you may input or enter content into the file. In edit mode you can move around the file, perform actions such as deleting, copying, search and replace, saving etc. vi always starts in Edit or Command mode.

We run vi with a single command line argument which is the file you would like to edit.

```
vi file.txt
```

Now we will edit a written file.

```
vi sample.txt
```

When you run this command it opens up the file. If the file does not exist then it will create it for you then open it up. (no need to touch files before editing them)

You always start off in edit mode so the first thing we are going to do is switch to insert mode by pressing i. You can tell when you are in insert mode as the bottom left corner will tell you. Now type in a few lines of text and press [Esc] which will take you back to edit mode.

## Saving and Exiting

:x! - Save and exit

:q! - discard all changes, since the last save, and exit

:w - save file but don't exit

:wq - again, save and exit

ZZ (Note: capitals) - again, save and exit

Most commands within vi are executed as soon as you press a sequence of keys. Any command beginning with a colon (:) requires you to hit to complete the command.

Save and exit the file you currently have open.

```
:x!
```

## Navigating a File

Go back into the file we just created and enter some more content.

```
vi sample.txt
```

In insert mode you may use the arrow keys to move the cursor around. Enter two more paragraphs of content then hit Esc to go back to edit mode.

Here are some of the many commands you may enter to move around the file. Let's have a play with them and see how they work.

Arrow keys - move the cursor around

j, k, h, l - move the cursor down, up, left and right (similar to the arrow keys)

^ (caret) - move cursor to beginning of current line

\$ - move cursor to end of the current line

nG - move to the nth line (eg 5G moves to 5th line)

G - move to the last line

w - move to the beginning of the next word

nw - move forward n word (eg 2w moves two words forwards)

b - move to the beginning of the previous word

nb - move back n word

{ - move backward one paragraph

} - move forward one paragraph

```
k
```

## Deleting

Here are some of the many ways in which you can delete content within vi.

x - delete a single character

nx - delete n characters (eg 5x deletes five characters)

dd - delete the current line

dn - d followed by a movement command. Delete to where the movement command would have taken you. (eg d5w means delete 5 words)

## Undo

u - Undo the last action (you may keep pressing u to keep undoing)

U (Note: capital) - Undo all changes to the current line

## Line Numbers

You may want to have line numbers in your doc, especially when editing code.

```
:set nu
```