Vi Text Editor

short line

## Vi Text Editor!

## Text editing, the right way.

## 

## **Introduction**

Master the Vi text editor and learn how to make complex edits on your files with less time and effort.

In the last section we created a few files but they were blank. A little boring but we have to start somewhere. 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 realise it is actually quite powerful. It's kinda like touch typing, initially learning is awkward and you wonder why you're bothering but once you get the hang of it you will not want to go back.

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.

This section and the next few sections are actually forming the foundation for the last few sections where we will put them all together and start doing some really funky stuff. I've chosen to look at Vi first so that your mind has a bit of time to process and make sense of it in preparation for later when we'll need it.

Vi is a very powerful tool. In this section my aim is not to cover everything that Vi can do but to get you up and running with the basics. At the end of the section I'll provide some links to resources where you can learn Vi further. I highly recommend you look into a few of them.

## A Command Line Editor

Vi is a command line text editor. As you would be quite aware now, the command line is quite a different environment to your GUI. It's a single window with text input and output only. Vi has been designed to work within these limitations and many would argue, is actually quite powerful as a result. Vi is intended as a plain text editor (similar to Notepad on Windows, or Textedit on Mac) as opposed to a word processing suite such as Word or Pages. It does, however have a lot more power compared to Notepad or Textedit.

As a result you have to ditch the mouse. Everything in Vi is done via the keyboard.

There are two modes in Vi. **Insert** (or Input) mode and **Edit** 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. A common mistake is to start entering commands without first going back into edit mode or to start typing input without first going into insert mode. If you do either of these it is generally easy to recover so don't worry too much.

When we run vi we normally issue it with a single command line argument which is the file you would like to edit.

**vi <file>**

If you forget to specify a file then there is a way to open a file within vi but it is easiest to just quit vi and have another go. Also remember that when we specify the file it can be with either an absolute or relative path.

Let's dive in and get started. It's going to be hard for me to demonstrate a lot of this so instead I'll list what I want you to type and you'll have to give it a go and see what happens.

First off let's move into the directory you created in the section on file manipulation. We're going to create a few files and this will keep them out of the way of your normal stuff.

Now we'll edit our first file.

1. vi firstfile

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) Once you enter vi it will look something like this (though depending on what system you are on it may look slightly different).

1. ~
2. ~
3. ~
4. ~
5. ~
6. "firstfile" [New File]

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.

1. ~
2. ~
3. ~
4. ~
5. ~
6. -- INSERT --

Now type in a few lines of text and press **Esc** which will take you back to edit mode.

## Saving and Exiting

There are a few ways to go about doing this. They all do essentially the same thing so pick whichever way you prefer. For all of these, make sure you are in edit mode first.

If you are unsure if you are in edit mode or not you can look at the bottom left corner. As long as it doesn't say **INSERT** you are fine. Alternatively you can just press **Esc** to be sure. If you are already in edit mode, pressing **Esc** does nothing so you won't do any harm.

* **ZZ** (Note: capitals) - 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

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 <enter> to complete the command.

Save and exit the file you currently have open

## Other ways to view files

vi allows us to edit files. If we wanted, we could use it to view files as well, but there are two other commands which are a bit more convenient for that purpose. The first one is **cat** which actually stands for concatenate. It's main purpose is to join files together but in it's most basic form it is useful for just viewing files.

**cat <file>**

If you run the command cat, giving it a single command line argument which is the file we just created, you will see it's contents displayed on the screen, followed by the prompt.

If you accidentally run **cat** without giving it a command line argument you will notice that the cursor moves to the next line and then nothing happens. Because we didn't specify a file, cat instead reads from something called STDIN (which we'll learn about in the section 'Piping and redirection' which defaults to the keyboard. If you type something then hit <enter> you will see cat mirror your input to the screen. To get out of here you may press **<Ctrl> + c** which is the universal signal for Cancel in Linux.

In fact, whenever you get in trouble you can generally press **<Ctrl> + c** to get yourself out of trouble.

1. cat firstfile
2. here you will see
3. whatever content you
4. entered in your file

This command is nice when we have a small file to view but if the file is large then most of the content will fly across the screen and we'll only see the last page of content. For larger files there is a better suited command which is **less**.

**less <file>**

less allows you to move up and down within a file using the arrow keys. You may go forward a whole page using the **SpaceBar** or back a page by pressing **b**. When you are done you can press **q** for quit.

Have a look at the file you just created now using both these commands.

## Navigating a file in Vi

Now let's go back into the file we just created and enter some more content. 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.

Below are some of the many commands you may enter to move around the file. 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 **n**th 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

If you type **:set nu** in edit mode within vi it will enable line numbers. I find that turning line numbers on makes working with files a lot easier.

## Deleting content

We just saw that if we want to move around in vi there are quite a few options available to us. Several of them also allow us to precede them with a number to move that many times. Deleting works similar to movement, in fact several delete commands allow us to incorporate a movement command to define what is going to be deleted.

Below are some of the many ways in which we may delete content within vi. Have a play with them now. (also check out the section below on undoing so that you can undo your deletes.)

* **x** - delete a single character
* **nx** - delete n characters (eg 5x deletes five characters)
* **dd** - delete the current line

## Undoing

Undoing changes in vi is fairly easy. It is the character **u**.

* **u** - Undo the last action (you may keep pressing u to keep undoing)
* **U (Note: capital)** - Undo all changes to the current line

## Taking it Further

We can now insert content into a file, move around the file, delete content and undo it then save and exit. You can now do basic editing in vi. This is just touching the surface of what vi can do however. I won't go into all the details here (I think I've thrown enough at you already) but I will give you a few things you may want to look into to further your expertise in vi. A basic search in your search engine of choice for **vi <insert concept here>** will find you many pages with useful information. There are many vi cheat sheets out there too which list all the commands available to you.

* copy and paste
* search and replace
* buffers
* markers
* ranges
* settings

Have fun and remember to keep at it. vi will be painful at first but with practice it will soon become your friend.

## **Summary**

**vi**

Edit a file.

**cat**

View a file.

**less**

Convenient for viewing large files.

**No mouse**

vi is a text editor where everything is done on the keyboard. If you can touch-type then this is great. If not then maybe you should think about learning.

**Edit commands**

There are many of them. Practice is the key to remember the most commonly used and useful ones.

## **Activities**

Let's play with some content.

* Start by creating a file and putting some content into it.
* Save the file and view it in both cat and less
* Go back into the file in vi and enter some more content.
* Move around the content using at least 6 different movement commands.
* Play about with several of the delete commands, especially the ones that incorporate a movement command. Remember you may undo your changes so you don't have to keep putting new content in.