# **Vim Editor Commands**

**Vim Editor Commands:**

1. The Vi editor has two modes: Command and Insert. When you first open a file with Vi, you are in Command mode. Command mode means you can use keyboard keys to navigate, delete, copy, paste, and do a number of other tasks—except entering text.
2. To enter Insert mode, press **i**. In Insert mode, you can enter text, use the Enter key to go to a new line, use the arrow keys to navigate text, and use vi as a free-form text editor. To return to Command mode, press the **Esc** key once.
3. In Vi's Command mode, almost every letter on the keyboard has a function.
4. To save a file, you must first be in Command mode. Press **Esc** to enter Command mode, and then type **:wq** to write and quit the file. The other, quicker option is to use the keyboard shortcut **ZZ** to write and quit. In Vi, write means save, and quit means exit. If you’ve made mistakes along the way while editing and want to back out (abandon) all non-saved changes, enter Command mode by pressing **Esc** and typing **:q**! This command quits without saving any changes and exits Vi.

Vi shortcuts:  
**$ vi <filename>** — Open or edit a file.  
**i** — Switch to Insert mode.  
**Esc** — Switch to Command mode.  
**:w** — Save and continue editing.  
**:wq or ZZ** — Save and quit/exit vi.  
**:q!** — Quit vi and do not save changes.  
**yy** — Yank (copy) a line of text.  
**p** — Paste a line of yanked text below the current line.  
**o** — Open a new line under the current line.  
**O** — Open a new line above the current line.  
**A** — Append to the end of the line.  
**a** — Append after the cursor’s current position.  
**I** — Insert text at the beginning of the current line.  
**b** — Go to the beginning of the word.

**e** — Go to the end of the word.

**x** — Delete a single character.

**dd** — Delete an entire line.

**Xdd** — Delete X number of lines.

**Xyy** — Yank X number of lines.

**G** — Go to the last line in a file.

**XG** — Go to line X in a file.

**gg** — Go to the first line in a file.

**:num** — Display the current line’s line number.

**h** — Move left one character.

**j** — Move down one line.

**k** — Move up one line.

**l** — Move right one character.

1. Search for next word (forward search). To search in vim, open up your file with vim **vim myfile.txt**, and press **ESC** to switch to normal mode. Type **/** followed by the word you are searching for. For example, if we want to search for **‘ERROR’** in our file. We type **/ERROR**. This will take us to the first occurrence of the word. To find the next occurrence, simply type **n**. And to go back to the previous occurrence, type **N**. To stop searching press **ESC** to go normal mode.
2. Search for previous word (backwards search). To search backwards in a file, open up the file, **vim myfile.txt**, and press **ESC** to switch to normal mode. Type **?** followed by the word. If we are searching for the word **‘INFO’** backwards, we type **?INFO**. Type **N** to search backwards and **n** to search forwards. To stop searching press **ESC** to go normal mode.
3. Case insensitive search. To ignore the case whilst searching, type **/** followed by a word, followed by **\c**. Let’s go through some examples, **/Linux** is case sensitive, **/Linux\C** is case sensitive, **/Linux\c** is case insensitive. It is also possible to set case sensitive search off in your vim config or the current file, by running **:set ignorecase** in normal mode.
4. Search for any line starting with a word. Example start with ‘def’: **/^def .**
5. Search for any line ending with a word. Example Ending with ‘return': **/return$ .**
6. Escaping special characters. Find ‘[0]’: **/\[0\]**