**TEXT EDITOR**

A text editor is a computer program that lets a user enter, change, store, and usually print [text](https://whatis.techtarget.com/definition/text) (characters and numbers, each encoded by the computer and its input and output devices, arranged to have meaning to users or to other programs). Typically, a text editor provides an "empty" display screen (or "scrollable page") with a fixed-line length and visible line numbers. You can then fill the lines in with text, line by line. A special command line lets you move to a new page, scroll forward or backward, make global changes in the document, save the document, and perform other actions. After saving a document, you can then print it or display it. Before printing or displaying it, you may be able to format it for some specific output device or class of output device. Text editors can be used to enter program language source statements or to create documents such as technical manuals.

A popular text editor in IBM's large or [mainframe](https://searchdatacenter.techtarget.com/definition/mainframe) computers is called XEDIT. In [UNIX](https://searchdatacenter.techtarget.com/definition/Unix) systems, the two most commonly used text editors are [Emacs](https://whatis.techtarget.com/definition/Emacs) and Vi . In personal computer systems, [word processor](https://searchwindowsserver.techtarget.com/definition/word-processor) s are more common than text editors. However, there are variations of mainframe and UNIX text editors that are provided for use on personal computers. An example is KEDIT, which is basically XEDIT for Windows.

**Examples of text editors**

* [**Notepad**](https://www.computerhope.com/jargon/n/notepad.htm) and [**WordPad**](https://www.computerhope.com/jargon/w/wordpad.htm) - Microsoft Windows included text editors.
* [**TextEdit**](https://www.computerhope.com/jargon/t/textedit.htm) - Apple computer text editor.
* [**Emacs**](https://www.computerhope.com/unix/uemacs.htm) - Text editor for all platforms that is a very powerful text editor once you've learned all its commands and options.
* [**Vi**](https://www.computerhope.com/unix/uvi.htm) and [**Vim**](https://www.computerhope.com/unix/vim.htm) - Other great editors primarily used with Linux but also available with multiple platforms.
* [**Word**](https://www.computerhope.com/jargon/m/microsoft-word.htm) - [Word processor](https://www.computerhope.com/jargon/w/word-processor.htm) for Windows and Apple computers.
* [**Writer**](https://www.computerhope.com/jargon/o/openoffi.htm) - Text editor and word processor.
* [**Atom**](https://www.computerhope.com/jargon/a/atom.htm) - Open source code and text editor.
* [**Ed**](https://www.computerhope.com/unix/ued.htm) - An ubiquitous file editor on Unix-like systems.
* [**Microsoft Edit**](https://www.computerhope.com/edithlp.htm) - MS-DOS text editor.
* [**NEdit**](https://www.computerhope.com/jargon/n/nedit.htm) - A graphical text editor.
* [**Pico**](https://www.computerhope.com/unix/upico.htm) - A console-based text editor.
* [**Notepad++**](https://www.computerhope.com/jargon/n/notepad-plus-plus.htm) - Our favorite free text editor, Notepad++ is a powerful alternative to Windows Notepad.
* [**TextPad**](https://www.computerhope.com/jargon/t/textpad.htm) - Our favorite [shareware](https://www.computerhope.com/jargon/s/sharewar.htm) text editor.

**VIM: -**

Vim editor is one of the more popular text editors we use today. It is a clone of the Vi editor and is written by Bram Moolenaar. It is cross platform editor and available on most popular platforms like Windows, Linux, Mac and other UNIX variants.

Vim is acronym for **Vi IM**proved. It is free and open source text editor written by Bram Moolenaar. It was first released in 1991 for UNIX variants and its main goal was to provide enhancement to the Vi editor, which was released way back in 1976.

Vim is considered as clone Vi editor. Like Vi, it is also command centric editor. One of the advantage of learning Vim is – it is available everywhere. Take any UNIX variant like Linux, Mac, HP-UX, AIX and many more, Vim is there by default. Vim traditionally does not have GUI but now there is separate installer called gVim which provides GUI.

## Features of Vim

This section discusses some of the important features of Vim −

* Its memory footprint is very low
* It is command centric. You can perform complex text related task with few commands
* It is highly configurable and uses simple text file to store its configuration
* There are many plug-in available for Vim. Its functionality can be extended in great manner using these plug-in
* It supports multiple windows. Using this feature screen can be split into multiple windows
* Same as multiple windows, it also supports multiple buffers
* It supports multiple tabs which allows to work on multiple files
* It supports recording features which allows to record and play Vim commands in repeated manner

## Installation on ubuntu

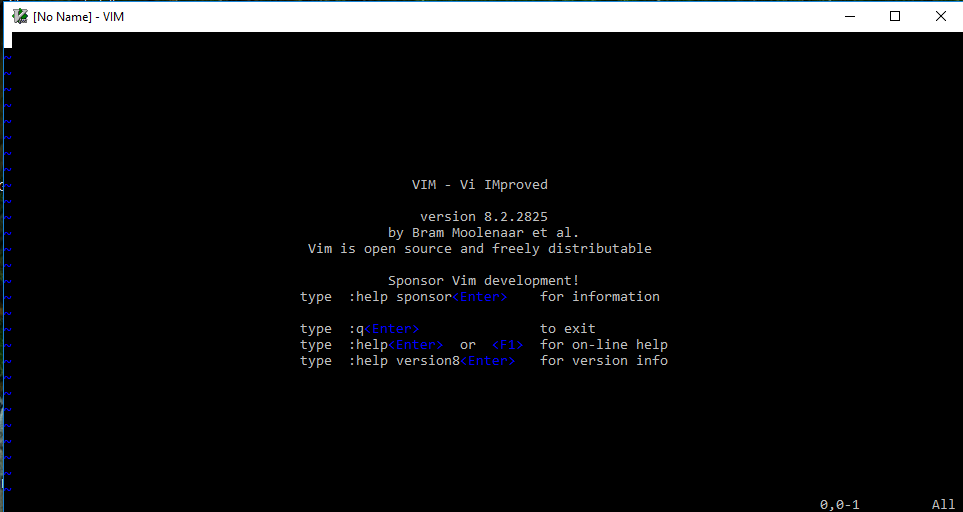
Installation on Linux platform is quite simple as compared to Windows. This section describes installation and configuration on ubuntu.

### **Installation**

* Execute below command in terminal −

$ sudo apt-get update

$ sudo apt-get install vim



**🡺Vim has three modes: -**

1. Command mode

2. Insert mode

3. Last line mode

**1.** **Command Mode: -**

When you first start editing a file with the vi editor you will be in vi command mode. In this mode you can issue many vi commands, including commands like insert, append, and delete, and other search and navigation commands that let you move around your file.

Possibly the most important thing to know is that when you're in command mode you can't insert text immediately. You first need to issue an insert, append, or open command to insert text. It is case sensitive.

**Move the cursor**

* h move one character left
* j move one row down
* k move one row up
* l move one character right

### **Text Deletion Commands**

* x Delete character
* dw Delete word from cursor on
* db Delete word backward
* dd Delete line
* d$ Delete to end of line
* d^ (d caret, not CTRL d) Delete to beginning of line

**Yank** (has most of the options of delete)-- VI's copy commmand

* yy yank current line
* y$ yank to end of current line from cursor
* yw (copy)yanks one word

**Paste** (used after delete or yank to recover lines.)

* p paste below cursor other command
* u - to undo the last the command and U to undo the whole line
* U - restore (undo) last changed line

**2.Insert mode: -**

Aside from command mode, the other mode you need to know about is insert mode, which allows you to insert text in Vi. Entering insert mode is easy once you know it exists — just press the ’i’ key once after you’ve positioned the cursor in command mode. Start typing and Vi will insert the characters you type into the file rather than trying to interpret them as commands. Once you’re done in insert mode, press the escape key to return to command mode.

**Text Entry Commands (Used to start text entry)**

* a - Append text following current cursor position
* A - Append text to the end of current line
* i - Insert text before the current cursor position
* I - Insert text at the beginning of the cursor line
* o - Open up a new line following the current line and add text there
* O - Open up a new line in front of the current line and add text there

**3**.**Last line mode: -**

The last vi mode is known as vi last line mode. You can only get to last line mode from command mode, and you get into last line mode by pressing the colon key (:)

After pressing this key, you'll see a colon character appear at the beginning of the last line of your vi editor window, and your cursor will be moved to that position. This indicates that vi is ready for you to type in a "last line command".

* To switch from command mode to this mode just type colon
* To switch from insert mode to this mode press Escape and type colon

**Command used in last line mode**

* :help [keyword] - Performs a search of help documentation for whatever keyword you enter
* :e [file] - Opens a file, where [file] is the name of the file you want opened
* :w - Saves the file you are working on
* :w [filename] - Allows you to save your file with the name you've defined
* :wq - Save your file and close Vim
* :q! - Quit without first saving the file you were working on