



# Introduction & Overview to VIM

**Last Updated:** 3/22/2019

## Introduction

VIM is a ubiquitous text editor that is configurable and great for creating or editing any text document. Vim was built to “Improve” the Unix text editor called VI on which it is based. VI was an inline command editor which was developed on a series of Unix command line editors. Both VIM and VI are modular editors, meaning you can perform different actions depending on the mode you are in.

## Installation

For installation instructions please follow the instructions on the VIM [download page](#)

## How to Start VIM

There are two commands that start the VIM environment. You can open a file you have already started developing by denoting the path to the file following the start command, or you can create a new text file by excluding any file path. For example ( `vim /path/to/file.txt` ) or ( `vim` )

*gvim file.txt*: The ( `gvim` ) command causes the editor to create a new window for editing.

*vim file.txt*: The editing occurs inside your command window

## VIM Modes

Both VIM and VI are modular editor and contain 6 different modes. There are four major modes (Command, Normal, Insert and Visual) necessary to start utilizing VIM. Normal mode is the default mode when starting VIM. From any other VIM mode, return to command mode by clicking ( `esc` ). Command mode is used to enter vim commands, to enter this mode, type a colon followed by the command ( `:<command>` ), for example ( `:quit` ).

Insert mode is one that people are most familiar with, this is the mode allows users to enter or change any content in your text document. To enter Insert command type ( *i* ) while in Command mode.

## General Mode Commands

### Normal Mode:

- *h* move one character left
- *j* move one row down
- *k* move one row up
- *l* move one character right
- *O* move to the beginning of the line
- *\$* move to the end of the line

### Insert Mode:

- *i* for 'insert', this immediately switches vim to insert mode
- *a* for 'append', this moves the cursor after the current character and enters insert mode
- *o* inserts a new line below the current line and enters insert mode on the new line
- *I* moves the cursor to the beginning of the line and enters insert mode
- *A* moves the cursor to the end of the line and enters insert mode
- *O* inserts a new line above the current one and enters insert mode on the new line

### Visual Mode

- Press ( *v* ) to enter visual mode, this will also mark a starting selection point
- You can now move the cursor to the desired end selection point; vim will provide a visual highlight of the text selection while you progress.
- *V* to enter visual line mode, this will make text selections by line
- block-visual: select any rectangular region. Enter by pressing ( *<ctrl>+v* )
- linewise-visual: always select full lines. Enter by pressing ( *<shift>+v* )

### FUN Resources

This is a really great source to practice your VIM in a fun [interactive game](#) and safe environment, away from any important text.

There is another great [source](#) as well.

## gVIM - Haha, it really is a thing

It is super ironical, but it stands for GUI VIM -- and sometimes its a useful thing. Especially when all that's needed is a quick look into a *really* big file.

