Reference 1 : <https://www.maketecheasier.com/start-with-vim-linux/>

> vim <filename.extension> // Opens a file if present else creates a file

Vim has two modes Edit mode, Visual / Command mode

> < press Esc> to enter Visual mode

> < press i > to enter Edit Mode

> < press v> to enter Visual Mode

In Command Mode, Command should begin with a (:) colon

> :q // exit, will ask you to save, if file not saved

> :q! // force exit, no save confirmation

> :w // save changes

> :wq // save and quit

> :

> Commands in Visual Mode

* x: to delete a character
* u: to undo an action (the equivalent of Ctrl+z)
* dd: to delete a line
* dw: to delete a word
* yy: to copy a line
* yw: to copy a word
* p: to paste the previously deleted or copied line or word
* e : to move to the next word (faster than just moving with the arrow keys)
* r: to replace a letter (press r, then the new letter)

And of course, there are more, but this is enough for now. If you master all of them, you will already be very fluent with Vim.

As a side note for those who always want more, you can type a number before any of these commands and the command will be executed that number of times. For example, 5x will delete five characters in a row, while 3p will paste three times.

## Design

Vim is designed around modality, composability and extensability.

Modality

Vim assumes that users edit text more often than they type it. It thus has a mode for inserting text and several modes for editing text. The behavior of vi depends on which mode it is set to **Command, Visual, Edit.**

### Composability

**Composability** is a system design principle that deals with the inter-relationships of components. A highly **composable** system provides components that can be selected and assembled in various combinations to satisfy specific user requirements.

Vim is designed similar to that of the Unix operating system. That is, functionality is provided through a library of simple commands that can be combined into more complex commands.

For example, the **w** command moves the cursor to the beginning of the next word. The **d** command is the deletion command. Thus **dw** deletes the next word; **d2w** deletes the next two words.

Extensability

Vim's functionality can be extended through a multitude of third-party plugins written in vimscript. The [vim-script](apt:vim-script) package includes a selection of useful plugins.

Managing vim plugins is unpleasant, but can be automated by plugin managers like [Vundle](https://github.com/gmarik/Vundle.vim).

**Extensibility** is a software engineering and systems design principle where the implementation takes future growth into consideration.

## Getting Started

To learn the basics of vim editing, issue the following command in a terminal emulator:

vimtutor

## Configuration

The ~/.vimrc file provides Vim's default configuration. To create and open the file for editing, run the following in a terminal:

vim ~/.vimrc

Below are examples of Vim configurations that can be added to the .vimrc file. Lines beginning with a **"** are comments:

" Indent automatically depending on filetype

filetype indent on

set autoindent

" Turn on line numbering. Turn it off with "set nonu"

set number

" Set syntax on

syntax on

" Case insensitive search

set ic

" Higlhight search

set hls

" Wrap text instead of being on one line

set lbr

" Change colorscheme from default to delek

colorscheme delek

Other .vimrc configurations can be found at [dotfiles](http://www.dotfiles.org/.vimrc). Note that some .vimrc files won't work if the correct plugins aren't installed.

Sources:

[Vimcasts](http://vimcasts.org/).

o[fficial documentation](http://vimdoc.sourceforge.net/)