

Vim Introduction

Vim is a text editor written by Bram Moolenaar in 1991. Based on the vi editor common to Unix-like systems, Vim is designed for use both from a command line interface and as a standalone graphical text editor.

Vim's basics are really very simple, but in combination the simple commands become very powerful. To open a file for editing with Vim, simply type `vim` in the terminal followed the name of the file. For example:

```
vim readMe.txt
```

If the file does not already exist, Vim will create a blank file.

Modes

For starters, you have **2 main modes**:

1. Command mode: all keystrokes are interpreted as commands
2. Insert mode: most keystrokes are inserted as text

To switch from the **insert mode** to the **command mode**, type `<Esc>`.

To switch from the **command mode** to the **insert mode** type one of:

- `i` ...switch to insert mode before the current position
- `a` ...switch to insert mode after the current position (append)
- `I` ...jump to the first non-blank character in the current line and switch to the insert mode
- `A` ...jump to the last character of the current line and switch to the insert mode

When in **insert mode**, you'll see `--INSERT--` at the bottom of the screen.

Movement

To navigate your cursor through the text, the simplest movement commands are:

- `h` ...move left
- `l` ...move right

- `k` ...move up
- `j` ...move down

Obviously these commands work only in the command mode. Alternatively, you can use the cursor arrow keys in both modes.

There are a lot of movement commands available in Vim that should make the arrow keys obsolete once you've memorized them. Practice makes perfect! Here the most important commands:

- `0` ...first column of the line
- `^` ...first non-blank character of the line
- `w` ...jump to next word
- `W` ...jump to next word, ignore punctuation
- `e` ...jump to word-end
- `E` ...jump to word-end, ignore punctuation
- `b` ...jump to word-beginning
- `B` ...jump to word-beginning, ignore punctuation
- `ge` ...jump to previous word-ending
- `gE` ...jump to previous word-ending, ignore punctuation
- `g_` ...jump to last non-blank character of the line
- `$` ...jump to the last character of the line

Editing

Inserting text is pretty simple in Vim, just type `i` to enter **insert mode** and start typing. There is also a set of editing commands from the **command mode**:

- `d` ...delete the characters from the cursor position up to the position given by the next command.
- `D` ...delete the current line.
- `x` ...delete the character under the cursor.
- `X` ...delete the character before the cursor (Backspace).
- `y` ...copy the characters from the current cursor position up to the position indicated by the next command.
- `Y` ...copy the current line.
- `p` ...paste previous deleted or yanked (copied) text after the current cursor position.

- `P` ...paste previous deleted or yanked (copied) text before the current cursor position.
- `.` ...repeat the last insertion or editing command (x,d,p...).

Undo and Redo

Don't be afraid to try the various commands, you can undo almost anything by hitting `u` in the **command mode** – even undo is undoable using `Ctrl-r`.

Searching and Replacing

Searching in Vim is very easy. Type `/` in the command mode and enter the term you want to search for, and Vim will search the file (in forward direction) for the term and highlight all results. Use `n` or `N` to cycle through the search results. To remove the highlighting, type the command `:noh`.

To replace one expression “old” with some other text “new”, type the command `:%s/old/new/gc`

This command substitutes occurrences of the word “old” with “new”. The two characters `gc` at the end are modifiers: the command looks globally for all occurrences within the file (`g`) and confirms if it really should replace each one shown (`c`).

Exiting and Saving

Once you're done editing and are ready to exit, hit `<ESC>` to enter **command mode** and use one of the following:

- `:w` ...write the file to disk (save).
- `:q` ...quit vim (a warning appears if a modified file has not been saved).
- `:q!` ...quit vim (with no warning).
- `:wq` ...write the file to disk and quit the editor.