1. multiple rows edit

ctl+v -> choose area -> shift+(i|a) -> edit -> edit

2. Vim behaves differently, depending on the name of command:

vim

the "normal" way, everything is default

ex

start in Ex mode. go to normal mode with the ":vi" command. Can also be done with the "-e" argument.

... more to refer `man vim`

OPTIONS

-e

Start Vim in Ex mode, just like the executable was called "ex"

vim's configuring

find and access into hidden file ".vim" in HOME directory, then create a file called "vimrc" and write setting inside it, such as:

set hlsearch

set autoindent

...

write following setting in vimrc, to settle the error of displaying CHINESE's characters

set fileencodings=utf-8,ucs-bom,gb18030,gbk,gb2312,cp936

set termencoding=utf-8

set encoding=utf-8

RE in vim

we could use RE in vim by:

/RE

:[n1,n2]x/RE/RE/x # x represent various command i.e. s, g, ...

three mode

general\_mode : press "Esc" back to the general\_mode . # in general\_mod:

nG "n" represent a number , move cursor to n row

/word search a character\_string "word" following after current cursor

/word1|word2|word3|... search character\_strings "word1,word2, word3 ... at one time, following after current cursor

?word search a character\_string "word" ahead before current cursor

n "n" is a English-letter , could repeat previous search action

N "N" is a English-letter , could repeat previous search action reversely

:n1,n2s/word1/word2/g "n1" and "n2" are both line number, could search character\_string "word1" between line n1 and n2 , then instead of word2

:1,$s/word1/word2/g same with previous line , but search through begin to end

:1,$s/word1/word2/gc same with previous line , but need confirmation beforre replace

x,X x to delete a character backwards , X to delete a character forwards

nx "n" is number , could continuous delete n characters backwards

d to delete&cut , but have to use with the other character following?

d$ delete from where the cursor at to the final of present line

dd delete a whole line where the cursor at

ndd "n" is number , could delete the whole n line next to the line where the cursor at

d1G delete all data from the first line to the line where the cursor at

dG delete all data from the line where the cursor at to the final line

yy copy the whole line where the cursor at

nyy "n" is number , could copy the whole n line next to the line where the cursor at

y1G copy all data from first line to the line where the cursor at

yG copy all data from the line where cursor at to the final line

p,P p to paste data which been copied to the line that next to the line where the cursor at , P to paste to previous line of the line where the cursor at

j combine the next line with the line where the cursor at

u undo previous action

[ctrl]+r redo previous action

. its the decimal point , same with previous line

:set modifiable make a file modifiable

:set nomodifiable make a file unmodifiable

edit\_mode : input "i" to enter edit\_mode , in edit\_mode , input a character would show up before the character which the cursor is at . # in edit mode:

i,I i to insert at where the cursor at , I to insert at the first non-space character of the line where the cursor is at

r,R r to replace the character one time where the cursor exactly at , R to replace the character continuously where the cu rsor exactly at

a,A a to insert at the character next to character where the cursor is at , A to insert behind the last character at the line where the cursor at

o,O o to insert a new line next to present line , O to insert a new line previous to present line

command\_mode : input any of ":" "/" "?" character , into the command\_mode . # in command\_mode:

:w save the data you have modified

:q quit vim

:q! do not save the modified data and force quit , the exclamation-mark always mean "force" in vim

:wq save then quit

ZZ double upper-case "Z" , to save then quits

:w [filename] save as filename

:r [filename] add the filename's content follow after present row

:n1,n2 w [filename] save contens of n1 row to n2 row as filename

:!command temporarily quit vim and show the result of command-executing eg. ":! ls/home" --> then we could lookup the contents under /home

:set nu show row-number

:set nonu hide row-number

:set hlsearch high light the searching

visual block:

v to select characters while the cursor passby

V to select line whle the cursor passby

[ctrl]+v to select bolck in rectangle

y copy the selected part

d delete the selected part

muti-file edit:

:n edit next file

:N edit previous file

:files list all file which present opened by vim

multi-window:

:sp [filename] open a new vim window , if fill the filename then the new vim window show the filename's content or would show the present file's content , sync-show

[ctrl]+w+down-arrow move cursor to the lower window , the correct press method is press the [ctrl] and w together then let go , then press down-arrow

[ctrl]+w+up-arrow same with previous line , but move cursor to the upper windowi

[ctrl]+w+left-arrow similar to previous

[ctrl]+w+right-arrow similar to previous, could be used in nerdtree existed

[ctrl]+w+q move to the window you wanna colse , then press q to quit

map kk :NERDTreeToggle<CR>

tell vim, when press "kk", execute command `NERDTreeToggle` and `enter` (<CR> represent `Enter`)

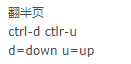
for more mapping info plz refer `help map` or `vi /media/root/qc/linux/note/map.txt`

<leader>

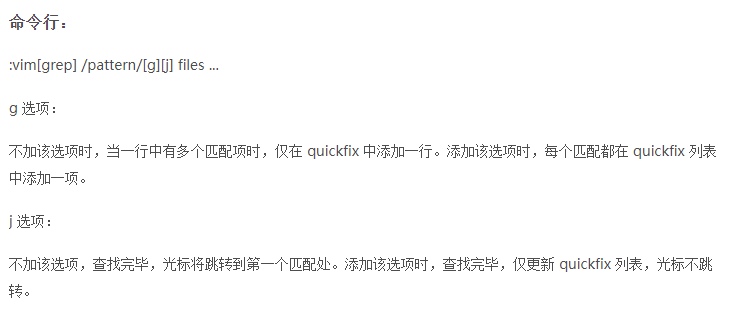
to define a mapping which uses the "mapleader" variable, the special string "<Leader>" can be used. It is replaced with the string value of "mapleader". If "mapleader" is not set or empty, a backslash is used instead.

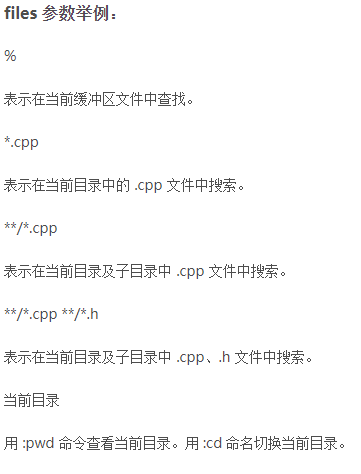
more info plz refer `:help <Leader>`

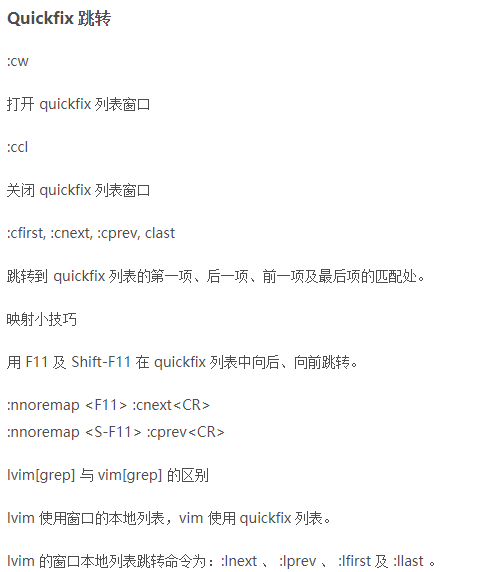
3.翻页



4.vimgrep







5. 查看历史命令

q: