

vi Commands

Entering vi

- vi filename - The filename can be the name of an existing file or the name of the file you want to create.
- view filename - Starts vi in "read only" mode. Allows you to look at a file without the risk of altering its contents.

Exiting vi

- :q - quit - if you have made any changes, vi will warn you of this, and you'll need to use one of the other quits.
- :w - write edit buffer to disk
- :w filename - write edit buffer to disk as filename
- :wq - write edit buffer to disk and quit
- ZZ - write edit buffer to disk and quit
- :q! - quit without writing edit buffer to disk

Positioning within text

By character

- left arrow - left one character
- right arrow - right one character
- backspace - left one character
- space - right one character
- h - left one character
- l - right one character

By word

- w - beginning of next word
- nw - beginning of nth next word
- b - back to previous word
- nb - back to nth previous word
- e - end of next word
- ne - end of nth next word

By line

down arrow	- down one line
up arrow	- up one line
j	- down one line
k	- up one line
+	- beginning of next line down
-	- beginning of previous line up
0	- first column of current line (zero)
^	- first character of current line
\$	- last character of current line

By block

(- beginning of sentence
)	- end of sentence
{	- beginning of paragraph
}	- end of paragraph

By screen

CTRL-f	- forward 1 screen
CTRL-b	- backward 1 screen
CTRL-d	- down 1/2 screen
CTRL-u	- up 1/2 screen
H	- top line on screen
M	- mid-screen
L	- last line on screen

Within file

nG	- line n within file
1G	- first line in file
G	- last line in file

Begin the vi editor exercises

Inserting text

a	- append text after cursor *
A	- append text at end of line *
i	- insert text before cursor *
I	- insert text at beginning of line *
o	- open a blank line after the current line for text input *
O	- open a blank line before the current

line for text input *

* Note: hit ESC (escape) key when finished inserting!

Continue the vi exercises

Deleting text

x	- delete character at cursor
dh	- delete character before cursor
nx	- delete n characters at cursor
dw	- delete next word
db	- delete previous word
dnw	- delete n words from cursor
dnb	- delete n words before cursor
d0	- delete to beginning of line
d\$	- delete to end of line
D	- delete to end of line
dd	- delete current line
d(- delete to beginning of sentence
d)	- delete to end of sentence
d{	- delete to beginning of paragraph
d}	- delete to end of paragraph
ndd	- delete n lines (start at current line)

Changing text

cw	- replace word with text *
cc	- replace line with text *
c0	- change to beginning of line *
c\$	- change to end of line *
C	- change to end of line *
c(- change to beginning of sentence *
c)	- change to end of sentence *
c{	- change to beginning of paragraph *
c}	- change to end of paragraph *
r	- overtype only 1 character
R	- overtype text until ESC is hit *
J	- join two lines

* Note: hit ESC (escape) key when finished changing!

Copying lines

```
yy          - "yank":  copy 1 line into buffer
nyy        - "yank":  copy n lines into buffer
p          - put contents of buffer after current
            line
P          - put contents of buffer before current
            line
```

Moving lines (cutting and pasting)

```
ndd        - delete n lines (placed in buffer)
p          - put contents of buffer after current
            line
P          - put contents of buffer before current
            line
```

Searching / Substituting

```
/str       - search forward for str
?str       - search backward for str
n          - find next occurrence of current string
N          - repeat previous search in reverse
            direction
```

The substitution command requires a line range specification. If it is omitted, the default is the current line only. The examples below show how to specify line ranges.

```
:s/old/new - substitute new for first occurrence
            of old in current line
:s/old/new/g - substitute new for all occurrences
            of old in current line
:1,10s/old/new - substitute new for first occurrence
            of old in lines 1 - 10
:.,$s/old/new - substitute new for first occurrence
            of old in remainder of file
:.,+5s/old/new - substitute new for first occurrence
            of old in current line and next 5 lines
:.,-5s/old/new - substitute new for first occurrence
            of old in current line and previous
            5 lines
:%s/old/new/g - substitute new for all occurrences
            of old in the entire file
```

```
:%s/old/new/gc - interactively substitute new for all
                  occurrences of old - will prompt for
                  y/n response for each substitution.
```

Miscellaneous commands

```
u                - undo the last command (including undo)
.                - repeat last command
xp              - swap two adjacent characters
m[a-z]          - set a marker (a - z)
'[a-z]          - go to a previously set marker (a - z)
:!command       - execute specified LINUX command
:r filename     - read/insert contents of filename after
                  current line.
:1,100!fmt      - reformat the first 100 lines
:!fmt          - reformat the entire file
```

vi Options

You can change the way vi operates by changing the value of certain options which control specific parts of the vi environment.

To set an option during a vi session, use one of the commands below as required by the option:

```
:set option_name
:set option_name=value
```

Some examples of the more common options are described below.

```
:set all        - shows all vi options in effect

:set ai         - set autoindent - automatically indents
                  each line of text

:set noai       - turn autoindent off

:set nu         - set line numbering on
```

```
:set nonu      - turn line numbering off

:set scroll=n   - sets number of lines to be scrolled
                to n.  Used by screen scroll commands.

:set sw=n      - set shiftwidth to n.  Used by autoindent
                option.

:set wm=n      - set wrapmargin to n.  Specifies number
                of spaces to leave on right edge of the
                screen before wrapping words to next
                line.

:set showmode  - reminds you when you are inserting
                text.

:set ic        - ignore case of characters when
                performing a search.
```

Options can be set permanently by putting them in a file called `.exrc` in your home directory. A sample `.exrc` file appears below. Note that you do not need the colon (`:`) as part of the option specification when you put the commands in a `.exrc` file. Also note that you can put them all on one line.

```
set nu ai wm=5 showmode ic
```