vi editor

- Visual Editor
- Started by Bill Joy in 1976 (co-founder of Sun Microsystems which is taken over by Oracle)
- Similar editors nano, ex, ed, kedit, gedit and more

vi editor characteristics

- Comes with every version of Linux/Unix
- Most popular editor among unix users and new version is called as “vim”
- vi is unformatted editor. No fonts, no graphics
- Case sensitive
- Multiple files editing   vi file1 file2 file3

vi editor starting

- Type vi <file name> at the shell prompt “$, # …”
- After pressing enter you will see blank screen with “~”character at the beginning of all the lines.
- “~” character indicates that line is blank

vi editor modes

1. **Command mode**

   - All the keys pressed by user are interpreted as editor commands.

   - Accessed by pressing “ESC”

2. **Insert mode**

   - permits insertion, deletion, replacement of text

   - Accessed by pressing “i”

3. **External command mode**

   - For executing regular linux commands

   - Accessed by in command mode “!<command>”

vi editor

Position by Character

Command	Function
h	Moves the cursor one character to left
Backspace key	Moves the cursor one character to left
l	Moves the cursor one character to right
Space bar key	Moves the cursor one character to right
0	Moves the cursor beginning of line
$	Moves the cursor end of line

vi editor

Position by Line

Command	Function
j	Moves the cursor one line down
k	Moves the cursor one line up
+	Moves the cursor one line down to beginning of next line
-	Moves the cursor one line up to beginning of previous line
Enter	Moves the cursor down to beginning of next line

vi editor

Position by Word

Command	Function
w	Moves the cursor right to first character of next word
e	Moves the cursor to end of current word
b	Moves the cursor back to first character of previous word

vi editor

Position in window

Command	Function
H	Moves the cursor first line on screen
M	Moves the cursor middle line on screen
L	Moves the cursor last line fo screen

vi editor

Scrolling

Command	Function
Ctrl f	Scrolls screen forward by full window
Ctrl b	Scrolls screen backward by full window

vi editor

Positioning on number line

Command	Function
G	Moves cursor to beginning of last line
nG	Moves cursor to beginning of nth line

vi editor

Commands for inserting text

Command	Function
a	Text input mode and appends after cursor
i	Text input mode and inserts at cursor position
A	Text input mode and appends at end of current line
I	Text input mode and inserts at beginning of current line
O	Text input mode and open new line above current line
o	Text input mode and open new line below current line
R	Text input mode and replaces from current cursor position

vi editor

Commands for deleting text

Command	Function
x	Delete character in current position
X	Delete character in left of curent position
dd	Delete current line
dw	Delete a word
Ndw, ndd	Deletes n words and n lines
d0	Deletes till beginning of line
d$	Deletes till end of line

vi editor

Commands for quitting vi

Command	Function
ZZ	Save and quit vi
!wq	Save and quit vi
!w filename	Save contents to new file
:w! filename	Overwrite existing file
:q!	Quit without saving
:q	Quit if changes are already saved

vi editor

Commands for search and replace

Command	Function
:s/str1/str2	Replace str1 with str2 in current line
:s/str1/str2/g	Replace str1 with str2 from current line till end
:m,n s/str1/str2/g	Replace str1 with str2 from m to n line
Nyy	Copy n lines
Nyyp	Copy n lines and paste.

vi editor

Commands for search and replace

Command	Function
:s/str1/str2	Replace str1 with str2 in current line
:s/str1/str2/g	Replace str1 with str2 from current line till end
:m,n s/str1/str2/g	Replace str1 with str2 from m to n line
Nyy	Copy n lines
Nyyp	Copy n lines and paste.

vi editor

Commands for search and replace

Command	Function
:set num	Display line numbers on
:set ic	Ignore case sensitive
:set mesg	Permit receipt of messages
:set show-mode	Show mode of operation
:set eb	Beep speaker

.exrc file stores these settings in home dir.

nano text editor - CHEAT SHEET

Editing Text

Cut Line	^K
Paste	^U
Justify Paragraph	^J
Replace	^\
Indent	Alt-{
Unindent	Alt-}

Moving in File

Goto Line	^_
Move Back One Character	^B
Move Forward One Character	^F
Move to Previous Line	^P
Move to Next Line	^N
Move to Beginning of Line	^A
Move to End of Line	^E

Exiting File

Close File	^X
Save	^O

Utilities

Display Cursor	^C
Help	^G
Search	^W
Spell Checker	^T

cheatography.com

http://harlankellaway.com/documents/nano_cheatsheet.pdf

Vi Editor Cheat Sheet

Movement Commands

Character

h, j, k, l	Left, down, up, right

Text

w, W, b, B	Forward, backward by word
e, E	End of word
(,)	Beginning of next, previous sentence
{, }	Beginning of next, previous paragraph
[[,]]	Beginning of next, previous section

Lines

0, $	First, last position of current line
^	First non-blank character of current line
+, -	First character of next, previous line
H	Top line of screen
M	Middle line of screen
L	Last line of screen
*n*H, *n*L	Line *n* from top, bottom of screen

Scrolling

[Ctrl]F, [Ctrl]B	Scroll forward, backward one screen
[Ctrl]D, [Ctrl]U	Scroll down, up one-half screen
[Ctrl]E, [Ctrl]Y	Show one more line at bottom, top of window
z[Enter]	Scroll until line with cursor is at top of screen
z.	Scroll until line with cursor is at middle of screen
z-	Scroll until line with cursor is at bottom of screen

Searches

/*pattern*	Search forward for *pattern*
?*pattern*	Search backward for *pattern*
n, N	Repeat last search in same, opposite direction
/, ?	Repeat previous search forward, backward
f*x*	search forward for character *x* in current line
F*x*	search backward for character *x* in current line
t*x*	search forward for character before *x* in current line
T*x*	search backward for character after *x* in current line
;	Repeat previous current-line search
,	Repeat previous current-line search in opposite direction

Line Number

[Ctrl]G	Display current line number
*n*G	Move to line number *n*
G	Move to last line in file
:*n*	move to line number *n*

Marking Position

m*x*	Mark current position as *x*
`*x*	Move cursor to *x*
``	Return to previous mark or context
'*x*	Move to beginning of line containing mark *x*
''	Return to beggining of line containing previous mark

Editing Commands

Insert

i, a	Insert text before, after cursor
I, A	Insert text at beginning, end of line
o, O	Open new line for text below, above cursor

Change

r	Replace with next typed characer

~	Change between uppercase and lowercase
c*m*	Change text block defined by movement command *m* (e.g., cw changes next word)
cc	Change current line
C	Change to end of line
R	Type over characters
s	Delete character and continue typing
S	Delete current line and continue typing

Delete, Move

x	Delete character
X	Delete character to the left of the cursor
d*m*	Delete text block defined by movement command *m* (e.g., dw deletes next word)
dd	Delete current line
D	Delete to end of line
p, **P**	Put deleted text before, after cursor
"*n***p**	Put text from delete buffer number *n* after cursor (for last nine deletions)

Yank (copy)

y*m*	Yank (copy) text block defined by movement command *m* (e.g., yw yanks next word)
yy, **Y**	Yank current line
"*a***yy**	Yank current line into named buffer *a*
p, **P**	Put yanked text before, after cursor
"*a***P**	Put text from buffer *a* before cursor

Other Commands

.	Repeat last edit command
u	Undo last edit
U	Undo changes to current line
J	Join two lines
[Ctrl]**L**, [Ctrl]**R**	Redraw screen

Invoking vi

vi *file*	Invoke vi editor on *file*
vi *file1 file2*	Invoke vi editor on files sequentially
view *file*	Invoke vi editor on *file* in read-only mode
vi -R *file*	Invoke vi editor on *file* in read-only mode
vi -r *file*	Recover *file* and recent edits after system crash
vi + *file*	Open *file* at last line
vi +*n* *file*	Open *file* at line number *n*
vi +/*pattern* *file*	Open *file* at *pattern*

Exit and Save Commands

ZZ	Save file and quit
:x	Save file and quit
:wq	Save ("write") file and quit
:w	Save file
:w!	Save file (overriding protection)
:30,60w *newfile*	Save lines 30 through 60 as file *newfile*
:30,60w>> *file*	Append lines 30 through 60 to file *file*
:w %.*new*	Save current buffer named *file* as *file.new*
:q	Quit
:q!	Quit, discarding any changes
Q	Quit vi and invoke ex
:e *file2*	Edit *file2* without leaving vi
:e! *file2*	Discard changes to current file, then edit *file2* without leaving vi
:n	Edit next file
:e!	Discard all changes since last save
:e#	Edit alternate file