

Press **ESC** (escape key) to get into the Vi command mode. You will need to press the **RETURN** key to execute a command starting with the character colon ":", slash "/", or question mark "?". Use **CTRL+c** to cancel a command, the dot "." to repeat your last command, and **!:cmd d** to execute a shell command named *cmd*.

Insert & Replace		insert mode
i	insert text before the cursor	
a	insert text after the cursor	
I	insert text at the beginning of the current line	
A	insert text at the end of the current line	
o	insert text in a new line below the cursor	
O	insert text in a new line above the cursor	
r	replace a character at the cursor position	
R	replace characters starting at the cursor position	

Cursor Motion		navigation
h or ←	move cursor left	
l or →	move cursor right	
k or ↑	move cursor up	
j or ↓	move cursor down	
G	goto the end of file	
nG or : <i>n</i>	goto the line number <i>n</i>	
0	move to the beginning of the line	
\$	move to the end of the line	
CTRL+f	move one screen view forward	
CTRL+b	move one screen view backward	

Going Further with:

:map for mapping a key in command mode to a group of commands. Example: **:map de :1,\$dM** will delete all lines when using the **:de** command.
:set to define or show your editor options. Example: **:set number** will make vi display line numbers, and **:set all** will display all current vi options.
:ab to define a text abbreviation in insert mode. Example: **:ab VIM Vi Improved** will auto-complete "VIM" in insert mode for the phrase "Vi Improved".

Ex Commands		vi features
:q	quit current open file	
:q!	force to quit without saving changes	
:w	save file	
:w file	save file as <i>file</i>	
:wq!	force to save file then quit	
:x,yw file	write from line <i>x</i> to line <i>y</i> into <i>file</i>	
:w >> file	append buffer to <i>file</i>	
:e file	edit another file	
:e! file	edit another file without saving current changes	
:r file	insert file content at the current cursor position	
:n	edit next file in vi arguments file list	
CTRL+G	display current file name and position	

You can start **vi** with arguments, the **-c** option is particularly useful to execute commands in a file from the command line. You can separate multiple commands with a pipe "|".

Example:
vi -c "%s/false/true/g|:wq" file.txt

Search & Substitute		patterns
/string	search forward for <i>string</i>	
?string	search backward for <i>string</i>	
n	repeat the last search	
:/s/str1/str2/gc	search and substitute <i>str1</i> by <i>str2</i> from the current line to the end of file. A line number can be specified before the comma ",". The "c" ask for confirmation before each substitution	
:%s/str1/str2/g	replace all <i>str1</i> by <i>str2</i> in all the file without prompting for confirmation	
:%s/str1/str2/	replace <i>str1</i> by <i>str2</i> on the first occurrence in each line of the file	

Undo, Delete, and Copy		edit commands
u	undo last change	
CTRL+r	redo last undo change	
M	set mark <i>A</i> to current position. Marks can be the letter [<i>a-z</i>] and [<i>A-Z</i>]	
y'A	yank (copy) from current line to mark <i>A</i>	
d'A	delete from current line to mark <i>A</i>	
P	put the buffer content before the cursor	
p	put the buffer content after the cursor	
x	delete character at cursor position	
dW	delete first word after cursor position	
d\$ or D	delete from cursor position to end of line	
dd	delete current line	
J	join current line with following line	