

Help

Command	vi	vim
Enter interactive <i>Help</i>		or <code>:help</code>
Get Help on a specific command		<code>:help command</code>
Follow a Help link		
Re-draw the screen		

Scrolling in a document

Command	vi	vim
Scroll screen <i>forward</i>		
Scroll screen <i>backward</i>		
Scroll half screen <i>down</i>		
Scroll half screen <i>up</i>		
Scroll one line <i>forward</i>		
Scroll one line <i>backward</i>		

Cursor movement

Command	vi	vim
Left		or
Down (<i>next</i> line)		or
Up (<i>previous</i> line)		or
Right		or
Beginning of line		
End of line		
Word forward		
Word <i>backward</i>		
Sentence forward		
Sentence <i>backward</i>		
Paragraph forward		
Paragraph <i>backward</i>		
Go to end of file		
Go to line number <i>x</i>	<i>x</i>	

Like most `vi` commands, movement can also be combined with a “repeat count” number. For example, `20)` moves the cursor 20 sentences forward while `4k25l` moves the cursor 4 lines up and 25 characters to the right.

Changing modes

Command	vi
<i>Escape</i> to command mode	
Enter <i>insert</i> mode before the cursor	
Enter insert mode at beginning of line	
Enter insert mode <i>after</i> the cursor	
Enter insert mode at end of line	
<i>Replace</i> a single character	
Enter overtyping (<i>replace</i>) insert mode	
<i>Replace/change</i> current word	
<i>Replace/change</i> current line	
<i>Replace/Change</i> to end of line	

Editing

Command	vi
<i>Undo</i> last edit command	
Restore current line to original state	
Repeat last edit command	
Delete (<i>x-out</i>) a single character	
<i>Delete</i> current word	
<i>Delete</i> current line	
<i>Delete</i> to end of line	
Copy (<i>yank</i>) current word	
Copy (<i>Yank</i>) current line	
<i>Yank</i> current word into buffer <i>x</i> (a-z)	<i>x</i>
<i>Yank</i> current line into buffer <i>x</i> (a-z)	<i>x</i>
<i>Paste</i> deleted/yanked text after cursor	
<i>Paste</i> deleted/yanked text before cursor	
<i>Paste</i> yanked text in buffer <i>x</i> (a-z)	<i>x</i>
<i>Paste</i> deleted in buffer <i>x</i> (1-9, it keeps track of the last 9 deletes)	<i>x</i>

Like most `vi` commands, movement can also be combined with a “repeat count” number. For example, `10dd` deletes the next 10 lines while `5yw` yanks

(copies) the next 5 words. Note that most commands act on the entire line when “doubled”, such as `dd` and `cc`. Note also that many commands act until the end of the line when capitalized, such as `C` and `D`.

Undo & fixing mistakes

Command	vi
<i>Undo</i> last edit command	
Restore current line to original state	
Repeat last edit command	

Marking positions

Command	vi
<i>Mark</i> current cursor position into <i>x</i>	<i>x</i>
Move cursor to marker <i>x</i>	<i>x</i>
Move cursor to last mark “touched”	
Move to start of line of last marker “touched”	
Move to start of line of marker <i>x</i>	<i>x</i>

Simple searching

Command	vi
Search forward	<i>pattern</i>
Find again forward	
Search backward	<i>pattern</i>
Find again backward	

See the section “Metacharacters” for information on patterns.

Search and replace

Command	vi
<i>Substitute</i> foo for bar on current line	<code>:s/foo/bar</code>
<i>Substitute globally</i>	<code>:s/foo/bar/g</code>
<i>Substitute globally</i> in the entire file	<code>:%s/foo/bar/g</code>
<i>Substitute</i> with <i>confirm</i>	<code>:s/foor/bar/c</code>
<i>Substitute global</i> with <i>confirm</i>	<code>:s/foo/bar/gc</code>
<i>Substitute global</i> in entire file with <i>confirm</i>	<code>:%s/foo/bar/gc</code>

