visual mode V visual line mode C-V visual block mode	gv select last selection	Paste before "*op system clipboard C-r* paste system clipboard (insert mode)	paste after "rop action into/from register r C-r" paste (insert mode)		# find word under cursor	N prev text	gg first line ?text find text C-b page zt screen on cursor	C-u 1/2 page		d delete/cut y yank c change (enter insert mode) shift left shift right indent swap case u lowercase u uppercase	dd delete line yy yank line cc change line < shift line left >> shift line right == indent line									
		9 prev x		B WORD (space)	$oldsymbol{gE}_{ ext{WORD (space) end}}$	enclosing (enclosing {	empty line	sentence						S delete char and inser	O insert line				
O line	∧ non-blank	Fx before x	Tx after x	b word	ge word end	h lefi	ZZ screen on curson	l right	e word end	W word	tx before x	f x find x	\$ line	I insert line start	insert	join with line (lines in visual)	a insert after	D delete to line end	delete to line end and insert	insert li
							j down		E WORD (space) end	W WORD (space)		next x			replace char	O insert line				
					sentence	} empty line	enclosing }	nclosing)							X delete char					
a select "a"n object INCLUD: surrounding spaces/tags W word	i select an "inner EXCLUDING sur spaces/tags W WORD p paragraph	r" object rrounding					zb screen on cursor													
S sentence	() 0 block { } {} block t html/xml t					C-d 1/2 page	C-f													
or " quoted string	t HIML/XML t	ag					/text	n next text	* find word under cursor	r										
↑ ↑ last location	last edit	# gg :#	0/0 matching bracket				G last line													

:g/pattern/d

	\n	toggle relative number				
	\d	dispatch and make current file				
	\W	toggle word wrap				
		Warre Carried and Ca Tr	Multiple cursors			
		YcmCompleter GoTo	C-n in Normal	highlights the current word under the cursor in Visual		
	F3	YcmCompleter GoToDeclaration		mode/finds next		
	F4	YcmCompleter GoToDefinition	C-n in Visual	virtual cursor at every line and leaves you in Normal mode.		
	F5	open nerd tree	C-p in Visual remove the current virtual cursor and go back to the			
	F8	toggle ctag bar	e p in visuui	previous virtual cursor location.	tmux	
			C-x in Visual	remove the current virtual cursor and skip to the next virtual	l C-a [enter text mode, hjkl to move, /? to search, C-ud to page
	tcomment	T. 1 ' 1'		cursor location.	C-a:	enter command mode
	gc {motion}	Toggle inline comments	:{r}MultipleCursorsFind	regex	C-a c	create window
	gc[Count]c{motion}	Toggle comment with count and motion	C/ 1		C-a.	rename window
	gcc	Toggle comment for the current line	Ctrlp	Cv. ID. Cl	C-a -	split horizontal
	g<{motion}	Uncomment region	C-p	CtrlP file	C-a	split vertical
	g	Uncomment the current line	\b	CtrlP buffer	C-a 19	jump to window
	g	Uncomment the current region as block	\m	CtrlP mixed	C-a 1	Jump to window
	g>{motion}	Comment region	\r	CtrlP most recently used	Buffer	
	g>c	Comment the current line	\t	CtrlP buffer tags	C-w ghjk	move to the left/bottom/top/right window
	g>b	Comment the current region as block	F5	(in CtrlP) purge the cache	C-w W	move to next window
	gc	Toggle comments (visual mode)	C-f/C-b	(in CtrlP) cycle between modes		horizontal split
	g>	Comment selected text (visual mode)	C-j/C-k	(in CtrlP) up/down	:sp	vertical split
	CTags		C-d	(in CtrlP) switch to filename only search	:vsp	•
	C-]	jump to tag	C-r	(in CtrlP) switch to regexp mode	.0 .h	open buffer with filename open buffer with number
Δ	C-t	ctags jump back from tag			.0 .bd	close current buffer
A insert line end	gd	goto local declaration	Errors		:DU	
	gD	goto global declaration	[q	:cprevious prev error	:DII	next buffer
	[g]	list matching tags]q	:cnext next error	:bp	prev buffer
	gC-]	jump to tag if defined once otherwise list matching tags	[Q	:cfirst first line	C-6	last buffer
	:tags	list tag stack]Q	:clast last line	:enew	new buffer, with filename creates
	:!ctags -Rc++-kinds=+pfields=+iaSlextra=+qlinks=yes.	update tag file	:copen	open Quickfix	:e	reload buffer, with filename creates
	\ut	update ctags	Format		:bufdo e	reload buffer all buffers, runs foreach buffer
	Cscope		\fx	format XML	A	
	:!findiname *.c -iname *.cc -iname *.hpp -iname *.h -iname *.cpp >.cscopelist.tmp; cscope -q -R -b -i .cscopelist.tmp; rm .cscopelist.tmp	o update csope db	\fj	format JSON	A	conitals to the least only one Cita
	C-\s	show uses of symbol in quickfix	:%!xmllintformat -	format xml in whole file	\a \	switch to the header/cpp file
	Find		:{range}!xmllintformat	- format xml in selection	\as	splits and switches
	\gf (Normal)	(global find) start empty Ag for cpp	:%!python -m json.tool	format json in selection	\av	vertical splits and switches
	\gf (Visual)	(global find) Ag with selection for cpp			\at	new tab and switches
	\f (Normal)	(find)start empty Ag in current file	V	enter Visual Mode	\an	cycles through next match
	\f (Visual)	(find) Ag with selection in current file	C-v	enter Visual Mode	E N/ - 4: -	
	ii (visuai)	(initia) Ag with selection in current inc	C-v	in insert mode adds tab char	Easy Motio	
	Replace		:retab	fix tabs to spaces	\\W	easy motion word forward
	rx	replace all chars in selection with x	:so ~/.vimrc	Reload vimrc	\\b	easy motion word back
	:{range}sort u	sort and remove dups	:! wc %	run external command on current file	\\k	easy motion up line
	:{range}sort	sort	:noh	turn off search highlight	\\J	easy motion down line
	:{range}sort!	sort reverse	vim -d file1 file2	diff 2 files		
	:{range}sort:	numeric sort	%	in {range} will use whole file		
	:s/foo/bar/g	Change each 'foo' to 'bar' in the current line.	:set pastetoggle	/		
		Change each 'foo' to 'bar' in all the lines., but ask for	:set paste	turn off paste formatting		
	:%s/foo/bar/gc	confirmation first.	:set nopaste			
		Change each 'foo' to 'bar' in all lines within a visual				
	:{range}s/foo/bar/gc	selection, but ask for confirmation first.				
	·a/nattarn/d	ramova avary line matching nattern				

remove every line matching pattern