

-down-motions								
	ts	sw	sts	et		tabstop	ts	Columns per tabstop
use spaces only	n	n	n	on		shiftwidth	sw	Columns per <<
use tabs only	n	n	0	off		softtabstop	sts	Spaces per tab
Set n to desired tab	widt	h (de	efaul	t 8)		expandtab	et	<tab> inserts spaces</tab>
MIXING TAI	BS	A	ND		SP	ACES I	S (th	RIGHT OUT. nat means don't do it.)
:retak			eplac bsto			•	s acc	cording to current
fileformat ff		Tr	y cha	angii	ng t	his if your lin	e-en	dings are messed up
list		Di	splay	y wh	ites	pace visibly a	accoi	rding to listchars
								:h left-right-mo

	fileform	at	ff		Try chan	ging th	is if you	ır line-	endings a	re mes	sed up	
	list				Display v	whitesp	ace visi	bly acc	cording to	listcha	ırs	
										:h left	-right-m	oti
ha	next racter		end of word	e	beginning of next word	W	end of WORD	Ε	beginning of next WORD	W	end of line	4

p	paste after cursor	P	paste before cursor	^[return to Normal mode
u	undo	^r	redo	•	repeat
file-searching gf	find file under cursor in path and jump to it	dd	delete current line	УУ	yank current line
X	delete character after cursor	%	jump to matching paren	r	replace char under cursor
nG	jump to line <i>n</i>	↑ jump-motions	jump back	jump-motions	jump forward
ZZ	center screen on cursor	zt	align top of screen with cursor	zb	align bottom of screen with cursor
	auto-indent current line	< <	shift current line left by shiftwidth	>>	shift current line right by shiftwidth

Using ^ [to return to Normal mode lets you keep your fingers on the home row. It's even easier if you map Caps Lock to Control!

			:n insert.txt
	COOL	INSER	T MODE STUFF
ΛW	delete word before cursor	^u	delete line before cursor
^r <i>r</i>	insert the contents of register r	^r=	use the expression register (try ^r=5+10)
^t	increase line indent by shiftwidth	^d	decrease line indent by shiftwidth
^x^1	line completion	^n	find next completion suggestion according to complete
			:h cmdline.txt

COM	MAND-LI	NE MODE (DNLY
insert word under cursor cmdline-editing	^r^w	completion suggestions cmdline-completion	۸d

:h sub-replace-\=

edit using Normal mode cmdwin	insert word under cursor cmdline-editing	^r^w	completion suggestions cmdline-completion	
Put cnoremap %% <c-r>=exp</c-r>			you can type ‰ in Command-line	
Supply % as a range to	the :substitut	ce command to r	run it on every line in the file.	
:%s/Scribbl/Desig	n/	"Scribbled" -> '	"Designed"	
Specify the "g" flag to a	apply the substitu	ition to every mat	tch on a line.	
:s/[dla]//g		"badly" -> "by"	:h s_flags, :h /[]	
Vim supports many regular expression features.				
:s/k/ax/		"Mook" -> "Max	x" :h usr_27, :h /.	
Use \ instead of . if	you want to sear	ch across multipl	le lines.	
:%s/heat*Bungl	e/anto/	"Cheatsheet\nE	Bungler" -> "Cantor" :h /\	
Special escapes can be used to change the case of substitutions.				
:s_\(f\)_\U\1\E	· 	"foobar" -> "FC	OObar" :h sub-replace-special	
Use : global to perfor	m a command or	matching lines.		
:g/foobar/delete		Delete all lines	containing "foobar"	
If your pattern contains	slashes, just use	e a different chara	acter as your delimiter.	
:s_Data/Lore_Bren	t Spiner_			
Use \= to evaluate expressions with replacement groups.				

"10 25" -> "21 36"

:h cmd	Normal mode <i>cmd</i> help
:hi_cmd	Insert mode <i>cmd</i> help
:h v_cmd	Visual mode <i>cmd</i> help
:hc_cmd	Command-line editing cmd help
:h:cmd	Command-line <i>cmd</i> help
:h 'option'	<i>Option</i> help
:helpgrep	Search through all help docs!



match; else list matching tags

			:h keycodes
(CR)	^m	\r	Enter
<tab></tab>	^i	\t	Tab
(C-n>	^n		Ctrl-n
<m-n></m-n>			Alt- <i>n</i>
(Esc)	^[Escape
<bs></bs>	^ h	\b	Backspace
			Delete

7 words :h word-motions http://www.vimcheatsheet.com 1 WORD

	:h optio
:set opt?	View current value of opt
:set noopt	Turn off flag opt
:set opt	Turn on flag opt
:set $opt=val$	Overwrite value of opt
:set $opt += val$	Append to value of opt
:echo &opt	Access opt as a variable

	:h buffers
:1s	List all open files
:b path	Jump to unique file matching path . Use <tab></tab> to scroll through available completions!
: b <i>n</i>	Jump to file <i>n</i> , number from first column of :ls
:bnext	Jump to next file
bprev	Jump to previous file
:bdelete	Remove file from the buffer list
:edit	Open a file for editing
:enew	Open a blank new file for editing
	:h windows
:split	Split current window horizontally
:vsplit	Split current window vertically
^w hjkl	Move cursor to window left, below, above or to the right of the current window
^w HJKL	Move current window to left, bottom, top, or right of screen
^w HJKL ^w r	Move current window to left, bottom, top, or
	Move current window to left, bottom, top, or right of screen
∧w r	Move current window to left, bottom, top, or right of screen Rotate windows clockwise
^w r ^w +-<>	Move current window to left, bottom, top, or right of screen Rotate windows clockwise Increase/decrease current window height/width

hidden	hid	Lets you switch buffers without saving
laststatus	ls	Show status line never (0), always (2) or with 2+ windows (1)
hlsearch	hls	Highlight search matches. Also see 'highlight'
number	nu	Show line numbers
showcmd	sc	Show commands as you type them
ruler	ru	Show line and column number of the cursor
backspace	bs	Set to '2' to make backspace work like sane editors
wrap		Control line wrapping
background	bg	Set to 'dark' if you have a dark color scheme

object motions to include delimiters or surrounding whitespace. For example, "(foo)" into "()", but da(will delete the parentheses as Use : map to

view all current

custom key

beginning text-

REGISTERS are CLIPBOARDS

All commands that delete, copy, or paste text use registers. To change which register is used by a command, type the register before the command. The default register is called "the unnamed register", and it is invoked with a pair of double-quotes (""). Typing dd or yy is the same as typing ""dd or ""yy. Think of the first " as a short way of saving "register", so "" is pronounced "register "", and "a, "register a".

of saying "register", s	o iii is pronounced	"register", and "a, "register a". :h registers
regis	ters	View all current registers
:echo	@	Access register r as a variable
11/	Last search pattern register	Contains the last pattern you searched for
	The black hole register	Use this to delete without clobbering any register ("_dd)
"0	Last yank register	Contains the last text you yanked
11	Last big delete register	Contains the last line(s) you deleted
"2-"9	Big delete register stack	Every time "1 is written to, its content is pushed to "2, then "2 to "3, and so on
	Small delete register	Contains the last text you deleted within a single line
"+	System clipboard	If the OS integration gods smile upon you, this register reads and writes to your system clipboard.
"a-"z	Named registers	26 registers for you to play with
"A-"Z	Append registers	Using upper-case to refer to a register will append to it rather than overwrite it

hitting q again

useful with a count

Record

Playback

Repeat last

playback

mappings. Read h map-whichs for a guide on which keys are best for your own custom mappings. Get used to Vim's help system it's a fantastic

Eric Sporkin M. Adam Price Daniel Hahler Bill LaPierre Laurence Emms @andersosthus Travis McHenry Rouven Hernier Digital Gnomes Sam Napolitano Jens O. Meiert Philip Ratzsch Hurricane Labs Michael Schmidt Matthew Datcher Arya Reais-Parsi David Macfarlane Christer Edvartsen St. Louis Vim Geeks Haraldur Tristan Gunnarsson and others!

vim one-liner used to sort the list of names by length: :exe 'g/^/let @x = len(getline(".")) | normal "xPa ' | sort n | :g//normal dw

Execute the contents of register **r**

Repeat the last @r, this is particularly

Record into register **r**. Stop recording by