|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Caster: Quick Reference (for v0.5.3)**  David Conway (CC Attribution 4.0 International)  **Numbers (1)**   |  |  | | --- | --- | | numb <0..1m>\*  word number <0..9>\* | print digits  print as words |   \* as of version 0.5.3, the special numbers from table (14) are available  **Alphabet, Punctuation, Spacing (2)**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | arch | brov | char | delta | echo  foxy | goof | hotel | India | julia  kilo | Lima | Mike | Novakeen  oscar | prime | Quebec | Romeo  Sierra | tango | uniform | victor  whiskey | x-ray | yankee | Zulu | | | letters | | | clamor  at sign  hashtag  dolly  carrot  starling  prekris\*  brax\*  curly\*  angle\*  quotes\*  thin quotes\*  boom\*\*  questo  period | dot  clear\*\*\*  deli\*\*\*  shock\*\*\* | !  @  #  $  ^  \*  ()  []  {}  <>  “”  ‘’  ,  ?  .  bkspc  del  enter | underscore  equals  modulo  slash  back slash  semper  deckle  ace\*\*\*  tabby\*\*\*  pipe sim  apostrophe  plus  minus  sauce\*\*\* ^  dunce\*\*\* ^  lease\*\*\* ^  ross\*\*\* ^ | | \_  =  %  /  \  ;  :  space  tab  |  ‘  +  -  up  down  left  right |   \* left key is pressed after  \*\* space bar is pressed after  \*\*\* repeatable by saying number after  ^ available as asynchronous auto-repeating  **Comparisons (3)**   |  |  | | --- | --- | | [is] greater than  [is] less than  [is] greater [than] [or] equal [to]  [is] less [than] [or] equal [to]  [is] equal to | >  <  >=  <=  == |   **Selection, Editing (4)**   |  |  | | --- | --- | | shackle  shin\* ^  queue\* ^  fly\* ^  stoosh\*\*  cut\*\*  spark\*\*  garb\*\*  drop\*\*  duple\*  cancel  save | select line  shift- <dir>  ctrl- shift- <dir>  ctrl- <dir>  copy  cut  paste  double-click + copy  double-click + paste  duplicate line  escape  ctrl-s |   \* takes optional number parameter for repetition  ^ takes optional direction parameter, one of the direction keys from table (2)  \*\* takes optional number parameter, which is used to determine a persistent clipboard slot  **Text Formatting (5)**   |  |  | | --- | --- | | Capitalization | | | yell\*  tie\*  gerrish\* ^  sing\*  laws\* | SOME WORDS  Some Words  someWords  Some words  some words | | Spacing | | | gum | gun\*  spine\*  snake\* ^ | somewords  some-words  some\_words |   \* capitalization and spacing can be combined into a single command; whether combining or not, they must be suffixed by “bow” and then dictation to format (ex: “tie bow some words”)  ^ gerrish defaults to gum; snake to laws  **Programming Core\* (6)**   |  |  | | --- | --- | | iffae  elsay  el iffae  switch  case of  breaker  default  for each  for loop  do loop  while loop  convert to integer  convert to floating-point  convert to string  lodge and  lodge or  lodge not  print to console  function  class  add comment  long comment  value not  return  value true  value false | if  else  else if  switch  case  break  default  for-each loop  for-i loop  do-while loop  while loop  logical and  logical or  logical not  system out  new function  new class  multi-line  null value |   \* must say “enable <language>” before use  **Mouse Replacement (7)**   |  |  | | --- | --- | | kick  kick mid  psychic  shift right-click  colic  curse  douglas  rainbow  legion  left point  center point  right point | left-click  middle-click  right-click  ctrl-click  move cursor by pixels  move cursor on grid  alternate grid  smart text selection  mouse to screen left  mouse to screen center  mouse to screen right |   **Template -Based Navigation (8)**   |  |  | | --- | --- | | jump in\*  jump out\*  jump back\*  fill <target>\* | move cursor inside next ([{<  move cursor past next )]}>  move cursor inside prev )]}>  highlight <target> in line |   \* all of these are asynchronous, and can be canceled with the word “cancel” if the search is taking too long; <target> only uses certain words; see *navigation.py* for the full list  **Window Management (9)**   |  |  | | --- | --- | | minimize  maximize  remax  dredge\* | minimize window  maximize window  force maximize  alt + tab |   \* slated for expansion  **Repetition (10)**   |  |  | | --- | --- | | again do  again <n> times  record from history | repeat last dictation  macro recorder |   **Aliasing (11)**   |  |  | | --- | --- | | alias <dict>  chain alias | alias highlighted text  alias as CCR |   **Passwords\* (12)**   |  |  | | --- | --- | | hash password <dict>  get password <dict> | hash dictation  use password seed |   \* dictation must be three words  **Other Miscellaneous Functionality (13)**   |  |  | | --- | --- | | Tau (kapa) sink  tau (kapa) dock  kraken  pita <text>\* | end + enter  end + semicolon + enter  ctrl + space  fuzzy string matching |   **Special Number Words\* (14)**   |  |  | | --- | --- | | zero  ane  twain  traio  fairn  faif  six  seven  eigen  nine | 0  1  2  3  4  5  6  7  8  9 |   \* commands which specify a number use these if the settings file is configured for it; all double-digits and larger are normal  Vim Commands mode (enter with “bitter butters”)   |  |  | | --- | --- | |  |  | |  |  | |  |  | | Tab edit | tabe~ | | Quit | q ~ | | Write and quit | wq ~ | | Show me the working directory | pwd~ | | Substitute | s/ | | Edit configuration file | tabe ~/.vimrc | | List all | !ls -al~ | | List in file | r!ls~ | |  |  | |  |  | |  |  | |  |  | |  |  |   Vim Navigation   |  |  | | --- | --- | | Up we go | gg | | Base camp | G | | Go to old | c-o | | Cursor top | s-h | | Cursor middle | s-m | | Cursor low | bottom | s-l | | Jump to line <number> | :<number>\n | | Search | \ | | Back search | ? | | Search this | \* | | Search this up | # |   Vim General   |  |  | | --- | --- | | Cancel | Escape, u |   Vim Tabs   |  |  | | --- | --- | | Glass next tab | g,t | | Glass previous tab | g,T |   Vim Windows   |  |  | | --- | --- | | Glass <direction> | c-w,<direction key> | | Glass split | c-w, s | | Glass vertical split | c-w, v |   Vim Normal mode commands   |  |  | | --- | --- | | Quick save | :,w | | ZZ | :,ZZ | |  |  | | [<n>] Pete | <n> . | | Go to definition (used in jedi vim) | \,d | | Shift left | << | | Shift right | >> | | Squirt | p | | Sprout | P | | Replace | r | | Shift replace | R | | yank line | y, y | | Yank inner <raip, rack, race, rangle> | y,i,<), ], }, >> | | Yank a <raip, rack, race, rangle> | y,a, <), ], }, >> | | Yank | y | | until [<n>] <letter> | t,<nth letter | | shift until [<n>] <letter> | T,<nth letter> | | Find [<n>] <letter> | f,<nth letter> | | Shift find [<n>] <letter> | F,<letter> | | Undo [<n>] |  | | Redo [<n>] |  | | Rip line [<n>] | Delete <n> lines | | Shift delete | s-d | | Delete inner <raip, rack, race, rangle> | d,i, <), ], }, >> | | Delete a <raip, rack, race, rangle> | d,a, <), ], }, >> | | Delete inner (word | whisky) | d,i,w | | Delete a (word | whisky) | d,a,w | | Delete | D | | Delete [<n>] word |  | | Join [<n>] |  | | Pete macro [<n>] | @,@ n times | | Single [<n>] | x n times | |  |  | | Change | c | | Change (whiskey | word) | c,w | | Change (echo | end) | c,e | | Change a paragraph | c,a,p | | Change inner paragraph | c,i,p | | Change a <raip, rack, race, rangle> | c,a,<> | | Change inner <raip, rack, race, rangle> | c,i,<> | | Shift change | C | | Change line | S | | Change chair | s | | (After | append) | a | | Shift (after |append) | A | | Oh | o | | Shift oh | O | |  |  | | Inner paragraph | i,p | | A paragraph | a,p | | Previous paragraph | { | | Next paragraph | } | | Text format | g,q | | Page center | z,. | | Echo [<n>] |  | | Whiskey [<n>] |  | | Brav [<n>] |  | | Previous | N | | Next | n | |  |  | | Visual block | c-v | | Visual line | s-v | | Visual | V | |  |  | | Lower case | g,u | | Upper case | g, U | | Swap case | ~ | |  |  | | Dolly | $ | | Modulo | % |   Insert mode commands   |  |  | | --- | --- | | (scratch|delete) [<n>] |  | | Slap |  | | Scratch|delete line |  | | Complete | c-j (for ultisnips) | | Snip next | c-l | | Snip previous | c-h | |  |  | |  |  |   Important vim notes:  For normal mode sequences you can say a sequence and then “repeat <n> times” to do it n times. |