

# Caster Quick Reference

## 1. Numbers

numb [minus]<numbers>[f] ..... **print digits**  
word numb <0...9> ..... **print as words**

## 2. Alphabet<sup>a</sup>

alpha	golf	mike	santa	yankee
bravo	hotel	nova	tango	zulu
carla	india	oscar	utah	
delta	julia	papa	vespa	
echo	kilo	quebec	whiskey	
foxy	lima	romeo	x-ray	

## 3. Punctuation<sup>d</sup>

tab	TAB	plus <sup>de</sup>	+
shin tab	... SHIFT-TAB	minus <sup>de</sup>	-
ace	SPACE	star   times <sup>d</sup>	*
quote <sup>f</sup>	"	divide <sup>d</sup>	/
chicky <sup>f</sup>   apostrophe	'	equals <sup>de</sup>	=
ticky <sup>f</sup>	`	bang	!
par <sup>bg</sup>	()	question	?
brax <sup>bg</sup>	[]	atty   at symbol	@
angle <sup>bg</sup>	<>	hashtag	#
curly <sup>bg</sup>	{ }	dollar	\$
comma <sup>ce</sup>	,	ampersand	&
dot   period <sup>c</sup>	.	mod <sup>de</sup>	%
slash	/	colon	:
backslash <sup>f</sup>	\	semicolon	;
underscore	_	carrot	^
dunder	--	tilde	~
pipe (sim   symbol)	..		

## 4. Comparisons

greater than <sup>d</sup>	>	less equal <sup>d</sup>	<=
less than <sup>d</sup>	<	equal to <sup>d</sup>	==
greater equal <sup>d</sup>	>=	not equal <sup>d</sup>	!=

<sup>a</sup>Prefix with "big" for capitals

<sup>b</sup>Left key is pressed after

<sup>c</sup>Space bar is pressed after

<sup>d</sup>Space inserted before and after

<sup>e</sup>Prefix with "short" to not insert space(s)

<sup>f</sup>Prefix with "double" to insert two of these characters

<sup>g</sup>Left or right available by prefixing with "open" or "close"

## 5. Directions

up <sup>d</sup>	up
down <sup>d</sup>	down
left <sup>d</sup>	left
right <sup>d</sup>	right

## 6. Editing

clear <sup>d</sup>	bkspc
deli <sup>d</sup>	del
enter <sup>d</sup>	enter
cancel	escape
save	ctrl - s

## 7. Selection, Movement

shackle	select line
shin <sup>a b</sup>	shift - <dir>
queue <sup>a b</sup>	ctrl - shift - <dir>
fly <sup>a b</sup>	ctrl - <dir>
splat <sup>a b</sup>	ctrl - del/bkspc
<sup>g</sup> home	home
<sup>g</sup> end	end
kraken	ctrl + space
hug <enclose> <sup>e</sup>	encloses target

## 8. Copy and Paste

copy <sup>c</sup>	copy
cut <sup>c</sup>	cut
paste <sup>c f</sup>	paste
grab <sup>c</sup>	double-click + copy
drop <sup>c</sup>	double-click + paste
duplicate <sup>b</sup>	duplicate line

<sup>d</sup>Repeatable by saying number after

<sup>a</sup>Takes optional direction parameter, one of the direction keys from table (5)

<sup>b</sup>Takes optional number parameter for repetition

<sup>c</sup>Takes par, brax, curly, angle, quote, chicky, ticky

<sup>g</sup>Prefix with "big" for ctrl-(home|end)

<sup>e</sup>Takes optional number parameter, which is used to determine a persistent clipboard slot

<sup>f</sup>Takes optional formatting parameters, e.g. "paste title snake"

## 9. Text Formatting

### Capitalisation

yell	SOME WORDS
title <sup>b</sup>	SomeWords
camel <sup>b</sup>	someWords
sing	Some words
laws	some words

### Spacing

ace (default)	some words
gum   gun	somewords
kebab	some-words
snake <sup>b</sup>	some_words
pebble	some.words
incline	some/words
dissent   descent	some\words

Capitalisation and spacing can be combined into a single command. They may be suffixed by "bow" and then dictation to format (ex: "title snake bow some words") to aid recognition.

Must pause after issuing command. This is not a CCR/merge command.

## 10. Free Dictation - Engine Formatting<sup>c</sup>

say	default engine formatting
cap	Initial letter capitalized
slip	initial letter lowercase

<sup>c</sup>Pass-through text formatting as defined by speech recognition engine.

<sup>b</sup>title and camel default to gum, snake to laws

## 11. Python (Visual Studio Code)

enable Python ..... **activate module**

bit shift (left right)	for
bitwise and .....	& from
bitwise or .....	function
bitwise complement .	~ global
bitwise xor .....	^ if
and	in
as	import
break	is
class	method
class method	none
comment	not
big comment	or
continue	pass
pie deli .....	del raise
derived class	return
elif	static method
else	true
except	try
false	while
finally	with
	short with

delete line	next tab <sup>a</sup>
expand <sup>a</sup>	previous tab <sup>a</sup>
shrink <sup>a</sup>	close tab <sup>a</sup>
fold [all]	split editor
unfold [all]	close editor
scroll [page] up <sup>a</sup>	next editor <sup>a</sup>
scroll [page] down <sup>a</sup>	previous editor <sup>a</sup>
hover	focus explorer
IntelliSense	focus editor
line <n>	terminal
(go peek side) definition	debug
peek references	debug continue
go symbol	debug stop
go bracket	breakpoint
show command palette	step (in out over) <sup>a</sup>
search for file	

## 12. C++ (Visual Studio)

enable C plus plus ..... **activate module**

TODO: List C++ com-   mands

TODO: List Visual Studio Helper Functions

## 13. Mouse Replacement

kick .....	left click
kick mid .....	middle click
psychic .....	right click
shift right click .....	shift right click
shift click .....	shift click
squat .....	left button down
bench .....	left button up
lean .....	right button down
hoist .....	right button up
colic .....	control left click
scree <sup>a</sup> .....	scroll wheel
curse .....	move cursor by pixels
douglas .....	move cursor on grid
rainbow .....	alternate grid
legion .....	smart text selection

## 14. Window Management

minimise win .....	minimise window
maximise win .....	maximise window
switch window .....	ctrl+alt+tab. Use <b>tab shin+tab, enter</b>
show windows (Dragon)	show list of windows
move window .....	move with arrow keys
window (left right) .	<b>snap window to left right</b>
send monitor (left right) .....	<b>send the current window to left/right monitor</b>
show work .....	<b>show active workspaces</b>
new work .....	<b>create a new workspace</b>
close work .....	<b>close the current workspace</b>
close all work .....	<b>close all workspaces</b>
previous work <sup>b</sup>	<b>go to the previous workspace</b>
next work <sup>b</sup> .....	<b>go to the next workspace</b>
go work <n> .....	<b>go to workspace number n</b>
send work <n> .	<b>send the current window to workspace n</b>
move work <n>	<b>move the current window to workspace n</b>

---

<sup>a</sup>Takes optional direction parameter, one of the direction keys from table (5)

<sup>b</sup>Takes an optional number parameter for repetition

## 15. Repetition

again do ..... **repeat last dictation**

again <n> times ..... **repeat last dictation\*n**

## 16. Dragon

(command | dictation | normal | spell | numbers)

mode on ..... **switch dragon modes (native dragon commands)**

scratch that ..... **undo previous dictation**

sratch <n> ..... **undo previous n dictations**

fix dragon double . **delete initial dictated char**