R3 CheatSheet - https://github.com/phreda4/ - PHREDA

Block construction					
(		Start block for IF or WHILE	)		End block for IF or WHILE
Nameless definition					
[		Start nameless definition	]		End nameless definitions
Control flow					
;		End of Word	EX	| v --	Run a word from address
Conditional					
0?	| a -- a	is TOS=Zero? conditional	1?	| a -- a	is TOS<>Zero? conditional
+?	| a -- a	is TOS>=0?	-?	| a -- a	is TOS<0?
<?	| a b -- a	is a<b? remove TOS	>?	| a b -- a	is a>b? remove TOS
=?	| a b -- a	is a=b? remove TOS	>=?	| a b -- a	is a>=b? remove TOS
<=?	| a b -- a	is a<=b? remove TOS	<>?	| a b -- a	is a<>b? remove TOS
AND?	| a b -- c	is a AND b? remove TOS	NAND?	| a b -- c	is a NAND b? remove TOS
BT?	| a b c -- a	is a<=b<=c? remove TOS			
Stack movements					
DUP	| a – aa	duplicate TOS	DROP	| a --	remove TOS
OVER	| ab -- aba	duplicate Second of Stack	PICK2	| abc -- abca	
PICK3	| abcd -- abcda		PICK4	| abcde -- abcdea	
SWAP	| ab -- ba		NIP	| ab -- b	
ROT	| abc -- bca		2DUP	| ab -- abab	
2DROP	| ab --		3DROP	| abc --	
4DROP	| abcd --		2OVER	| abcd -- abcdab	
2SWAP	| abcd -- cdab				
Return Stack					
>R	| a --	rstack: -- a	R>	| -- a	rstack: a --
R@	| -- a	rstack: a -- a			
Logic operators					
AND	| a b -- c	c=a AND b	OR	| a b -- c	c=a OR b
XOR	| a b -- c	c=a XOR b	NOT	| a -- b	b=NOT a
Aritmetic operators					
+	| a b -- c	d=a+b	-	| a b -- c	d=a-b
*	| a b -- c	d=a*b	/	| a b -- c	d=a/b
<<	| a b -- c	d=a shift left b	>>	| a b -- c	d=a shift rigth b
>>>	| a b -- c	d=a shift rigth b w/o sign	MOD	| a b -- c	d=a mod b
/MOD	| a b -- c d	c=a/b d=a mod b	*/	| a b c -- d	d=a*b/c
*>>	| a b c -- d	d=(a*b)>>c	<</	| a b c -- d	d=(a<<c)/b
NEG	| a -- b	b=-a	ABS	| a -- b	b=|a|
SQRT	| a -- b	b=square root(a)	CLZ	| a -- b	b=count lead zeros of a
Memory fetch and store					
@	| a -- [a]		C@	| a -- b[a]	
Q@	| a -- q[a]		@+	| a -- b [a]	
C@+	| a -- b b[a]		Q@+	| a -- b q[a]	
!	| a b --		C!	| a b --	
Q!	| a b --		!+	| a b -- c	
C!+	| a b -- c		Q!+	| a b -- c	
+!	| a b --		C+!	| a b --	
Q+!	| a b --				
Auxiliary registers					
>A	| a --	load register A	A>	| -- a	push register A
A@	| -- a		A!	| a --	
A+	| a --		A@+	| -- a	
A!+	| a --		>B	| a --	load register B
B>	| -- a	push register B	B@	| -- a	
B!	| a --		B+	| a --	
B@+	| -- a		B!+	| a --	
Memory copy and fill					
MOVE	| d s c --	copy S to D, C dword	MOVE>	| d s c --	copy from S to D, C dword in rev.
FILL	| d v c --	fill D, C dword with V	CMOVE	| d s c --	copy from S to D, C bytes

CMOVE>	| d s c --	copy S to D, C bytes in rev.	CFILL	| d v c --	fill from D, C bytes with V
QMOVE	| d s c --	copy S to D, C qwords	QMOVE>	| d s c --	copy from S to D, C qwords in rev.
QFILL	| d v c --	fill D, C qwords with V			
Operating System					
UPDATE	| --	update SO events	REDRAW	| --	refresh graphic buffer
MEM	| -- a	start memory free	VFRAME	| -- a	frame buffer adress
SH	| -- a	screen height	SW	| -- a	screen width
XYPEN	| -- x y	position of mouse or pen	BPEN	| -- a	key state of mouse or pen
KEY	| -- a	key code	CHAR	| -- a	character ascii code
TIME	| -- a	Hour(8):min(8):sec(8)	DATE	| -- a	Year(16):month(8):day(8)
MSEC	| -- a	milisecond of system	APPEND	| m cnt "fn" --	append file from M, C bytes
LOAD	| m "fn" -- lm	load file in M, last in LM	SAVE	| m cnt "fn" --	save file from M, C bytes
FFIRST	| "f" -- s	get first struct of folder "f"	FNEXT	| a -- s	next struct or 0 to end
SYS	| "sys" --	call SO to run program			
Graphics drawing					
INK	| -- color	value of pen color	'INK	| -- 'ink	adress of color to set
ALPHA	| a --	set alpha value	OP	| x y --	set last point
OPX	| -- opx	last x point	OPY	| -- opy	last y point
LINE	| x y --	lineto	CURVE	| x y x y --	curve cuadratic bezier
CURVE3	| x y x y x y --	curve qubic bezier	PLINE	| x y --	lineto polygon
PCURVE	| x y x y --	curve cuadratic bezier poly	PCURVE3	| x y x y x y --	curve qubic bezier polygon
POLI	| --	fill polygon			
Video Playback (r3v version)					
VIDEO	| "fn" w h --	0 close video	VIDEOSHOW	| w h -- v	
VIDEOSIZE	| w h --				
Sound and Music (r3v version)					
SLOAD	| "fn" -- s		SFREE	| s --	
MLOAD	| "fn" -- m		MFREE	| m --	
SPLAY	| s --	0 stop sound	MPLAY	| m --	0 stop music

Prefix	
:	define CODE, :: Export word
#	define DATA, ## Export word
^	Include source code in filename
'	Adress of word, code or data
|	Commento to end of the line
“	String to next “, “” for “ character
$	Hex numbers
%	Binary numbers, 0 can be .