		X	Y	Z
33	separator			
34	.' (transpose) / permute	rot90	system	full
35	for	repmat	repelem (run-length decoding)	blanks
36	specify outputs	display stack (debug)		fopen, fwrite, fclose
37	specify inputs		char(vpa(...))	fopen, fread, fclose
38	comment	class	cast	typecast
39	alternative default input/output spec	intersect	and	bitand
40	*Not used. String delimiter*		run-length encoding	now / clock
41	( ) assignment indexing / split	{ } assignment indexing	( ) assignment ind. with final : / split	( ) assignment ind. with initial : / split
42	( ) reference indexing	{ } reference indexing	( ) reference ind. with final :	( ) refererence ind. with initial :
43	.*	kron	matrix product	Cartesian product
44	+	conv	conv2	conv2(..., 'same')
45	,	cos	sin	tan
46	-	setdiff	deconv	
47	.	break	continue	pause	bitget
48	/	./	angle	matrix /	unwrap
49	0	*Not used*	predefined literals	predefined literals	
50	1	*Not used*	predefined literals	predefined literals	
51	2	*Not used*	predefined literals	predefined literals	
52	3	*Not used*	predefined literals	predefined literals	
53	4	*Not used*	predefined literals		
54	5	*Not used*	predefined literals		
55	6	*Not used*	predefined literals		
56	7	*Not used*	predefined literals		
57	8	*Not used*	predefined literals		
58	9	*Not used*	predefined literals		
59	:	colon (function)	linearize array	comma-separated list	bitset
60	;		acos	asin	atan2
61	<	<	min	cummin	
62	=	==	isequal	strcmp	strncmp
63	>	>	max	cummax	
64	?	if		why	sparse
65	@	push "for" value / "while" index	push "for" index	perms	randperm
66	A	all	all(..., 1)	dec2base. Larger base, any symbols	base2dec. Larger base, any symbols
67	B	logical(dec2bin(...)-'0')	bin2dec(char(...+'0'))	dec2bin	bin2dec
68	C		histcounts	im2col	im2col(..., 'distinct')
69	D	disp(num2str(..., ...))	disp(num2str(...))	sprintf / fprintf	disp
70	E	multiply by 2	replace elements in array		
71	F	*Not used. False (literal)*			
72	G	Paste from clipboard G (user-input)	plot	imwrite / imagesc / image / imshow	appearance of graphics / format
73	H	Paste from clipboard H	Copy to clipboard H		
74	I	Paste from clipboard I	Copy to clipboard I		
75	J	Paste from clipboard J	Copy to clipboard J		
76	K	Paste from clipboard K	Copy to clipboard K		
77	L	Paste from clipboard L (multi-level)	Copy to clipboard L (multi-level)	gallery	
78	M	Paste from clipboard M (function-input)	mode		
79	N	stack size	nchoosek (first input: array)	NaN	isnan
80	O	zeros	datestr	datenum	datevec
81	P	flip	flipud	pi	pdist2
82	Q	increment by 1	accumarray		polyval / roots
83	R	triu	triu(...,1)	tril	tril(...,-1)
84	S	sort	sortrows	circshift	sign
85	T	*Not used. True (literal)*		toeplitz	
86	U	str2num / string to array	str2double		
87	V	num2str			
88	W	2 raised to input			
89	X	*Not used*	regexp	regexprep	
90	Y	*Not used*		inf	isinf
91	Z	*Not used*			
92	[	*Not used. Array delimiter*	ind2sub		
93	\	mod	mod(...-1)+1	matrix \	
94	]	end (loops or conditional branches)	sub2ind		
95	^	.^	sqrt	matrix ^	Cartesian product
96	_	unary minus			
97	`	do...while	while	tic	toc
98	a	any	any(..., 1)	padarray	base2base
99	b	bubble		strsplit	
100	c	char (also for cell array)	cat	strcat	strjoin
101	d	diff	diag / spdiags	blkdiag	gcd
102	e	reshape / squeeze			exp
103	f	find	strfind	factor	
104	g	logical / cell2mat	ndgrid		gammaln
105	h	horzcat	{..., ...}	hankel	hypergeom
106	i	input	urlread	imread	
107	j	input(...,'s')	real	imag	conj
108	k	lower / floor	upper / ceil	closest values	
109	l	ones		log. With two inputs, specifies base	log2
110	m	ismember	ismember(...,'rows')	mean	lcm
111	n	numel	nchoosek (first input: numbers)	interp1	
112	o	double / cell array to numeric array	int64	round / change case	fix
113	p	prod	prod(..., 1, ...)	cumprod	isprime / totient function
114	q	decrement by 1	quantile	n-th prime / next prime	primes
115	r	rand	randn	randi	randsample
116	s	sum	sum(..., 1, ...)	cumsum	std
117	t	duplicate elements			strrep
118	u	unique	unique(...,'rows')		strjust
119	v	vertcat	remove all blanks	strtrim	deblank
120	w	swap			
121	x	delete from stack	clc		
122	y	duplicate element	eye	hypot	size
123	z	nnz	nonzeros		
124	{	*Not used. Cell array delimiter*	num2cell	mat2cell	mat2cell(x,ones(size(x,1),1),size(x,2))
125	|	abs / norm / determinant	union	or	bitor
126	}	else			split array
127	~	Not	setxor	xor	bitxor