

	X	Y	Z
separator			
! (transpose) / permute	rot90	system	full
for	repmat	repelem (run-length decoding)	blanks
# specify outputs	display stack (debug)	sound, soundsc, audiowrite	fopen, fwrite, fclose
\$ specify inputs		char(vpa(...))	fopen, fread, fclose
% comment	class	cast	typecast
& alternative default input/output spec	intersect	and	bitand
' Not used. String delimiter		run-length encoding	now / clock
() assignment indexing / split	{ } assignment indexing	() assignment ind. with final : / split	() assignment ind. with initial : / split
() reference indexing	{ } reference indexing	() reference ind. with final :	() reference ind. with initial :
* *	kron	matrix product	Cartesian product
+ +		conv2	conv2(..., 'same')
	cos	sin	tan
-	setdiff	deconv	
break	continue	pause	bitget
/	angle	matrix /	unwrap
0 Not used	predefined literals	predefined literals	
1 Not used	predefined literals	predefined literals	
2 Not used	predefined literals	predefined literals	
3 Not used	predefined literals	predefined literals	
4 Not used	predefined literals	predefined literals	
5 Not used	predefined literals	predefined literals	
6 Not used	predefined literals	predefined literals	
7 Not used	predefined literals	predefined literals	
8 Not used	predefined literals		
9 Not used	predefined literals		
:	colon (range)	comma-separated list	bitset
;		acos	atan2
<	min	cummin	
==	isequal	strcmp	
>	max	cummax	
? if		why	sparse
@ push "for" value / "while" index	push "for" index	perms	randperm
all	all(..., 1)	dec2base. Larger base, any symbols	base2dec. Larger base, any symbols
B logical(dec2bin(...)-'0')	bin2dec(char(...+'0'))	dec2bin	bin2dec
C	histcounts	im2col	im2col(..., 'distinct')
D disp(num2str(..., ...)) / mat2str	disp(num2str(...))	sprintf / fprintf	disp
E multiply by 2	replace elements in array		
F Not used. False (literal)		exponents of prime factorization	
G Paste from clipboard G (user-input)	plot	imwrite / imagesc / image / imshow	appearance of graphics / format
H Paste from clipboard H	Copy to clipboard H		
I Paste from clipboard I	Copy to clipboard I	col2im	image processing functions
J Paste from clipboard J	Copy to clipboard J		
K Paste from clipboard K	Copy to clipboard K		
L Paste from clipboard L (multi-level)	Copy to clipboard L (multi-level)	gallery	
M Paste from clipboard M (function-input)	mode		
N stack size	nchoosek (array)	NaN	isnan
O zeros	datestr	datenum	datevec
P flip	flipud	pi	pdist2
Q increment by 1	accumarray		polyval / roots / polyfit
R triu	triu(...,1) / build matrix	tril	tril(...,-1) / build matrix
S sort	sortrows	circshift	sign
T Not used. True (literal)		toeplitz	
U str2num / string to array / square	str2double		
V num2str			
W 2 raised to input			
X Not used	regexp	regexprep	
Y Not used		inf	isinf
Z Not used			
[Not used. Array delimiter	ind2sub		
\ mod	mod(...-1)+1	matrix \	divisors
] end (loops or conditional branches)	sub2ind		
^	sqrt	matrix ^	Cartesian power
~ unary minus			
- do...while	while	tic	toc
a any	any(..., 1)	padarray / unpad array	base2base
b bubble		strsplit	
c char (also for cell array)	cat	strcat	strjoin
d diff	diag / spdiags	blkdiag	gcd
e reshape / squeeze			exp
f find	strfind	factor	
g logical / cell2mat	ndgrid	gamma / gammaln / betainc	gammaln / betaln
h horzcat	{..., ...}	hankel	hypergeom
i input	uread	imread	
j input(...,'s')	real	imag	conj / real and imag
k lower / floor	upper / ceil	closest values	
l ones	clamp (limit to a range)	log. With two inputs, specifies base	log2
m ismember	ismember(...,'rows')	mean	lcm
n numel / size	nchoosek (numbers) / multinomial c.	poly / interp1	
o double / cell array to numeric / parity	int64	round / change case	fix
p prod	prod(..., 1, ...)	cumprod	isprime / totient function
q decrement by 1	quantile	n-th prime / next prime	primes
r rand	randn	randi	randsample
s sum	sum(..., 1, ...)	cumsum	std / cov
t duplicate elements			strrep
u unique	unique(...,'rows')		strjust
v vertcat		eig / svd / strtrim	deblank / symmetric range
w swap			
x delete from stack	clc		
y duplicate element	eye	hypot	size
z nnz	nonzeros / remove whitespace		
{ Not used. Cell array delimiter	num2cell	mat2cell	mat2cell(x,ones(size(x,1),1),size(x,2))
abs / norm / determinant	union	or	bitor
} else / finally			split array
~ Not	setxor	xor	bitxor