

### Basic

Help	Exit	Constants	Format	Type	ML
doc	quit	pi	compact	cast(b,'like',a)	A=TP+TN/T
help	exit	i	loose	char('a'+1)='b'	P=TP/TP+FP
demo		j (=i)	long (15 dp.)		
lookfor		inf ( $\infty$ )	short (4 dp.)		

### Randomizing

Real Numbers	Integers	Seeding
rand: range 0-1	round(rand*N): range 0-N	rng('shuffle'): sets clock as seed
rand*N: range 0-N	randi(n): range 1-n	rng(intseed): introduces seed
randi*(high-low)+low: range low-high	randi([min,max]): range min-max	rng('default'): default seed

### Numerical

Division	Round	Root	Logs	Degrees	Sign
rem(3,2)=1	fix	sqrt	log(x)=ln	deg2rad	sign(n)=1 if n>0
mod(3,2)=1	floor	nthroot(64,3)=4	log2(x)=base 2	rad2deg	sign(n)=0 if n=0
	ceil		log10(x)=base 10		sign(n)=-1 if n<0
	round		exp(n)=e <sup>n</sup>		

### Arrays

Creating	Referring
linspace(x,y,n): linear spacing, range x-y, n elem (def n=100)	mat(n): n <sup>th</sup> element
logspace(x,y,n): log spacing, range x-y, n elem (def n=50)	length(vec): total elem in vec
rand(n,m): n*m rand matrix	[r, c]=size(mat): r=rows, c=columns
randi([min,max],n,m): n*m int matrix	length(math): r/c length (biggest)
zeros(n,m): n*m matrix of 0	empty vec: [ ]
ones(n,m): n*m matrix of 1	numel(mat): total elem in mat

### Misc Matrix

diff	diff between consecutive vec	dot	dot product
any	if true for at least 1 elem	cross	cross product
all	if true for all elem	diag	returns diag of matrix
find	returns indices that meet criteria	trace	sum of elem in diag
isequal	compares arrays	eye	n*n identity matrix
isdiag	true if diag matrix	issymmetric	true if symmetric
mean(data,'omitnan')	ommits NaN to calc mean		

### Plotting

Basic	**Colors	Markers
-------	----------	---------



### Plotting (cont)

<code>plot(x ,y, 'r*')</code>	b: blue	.: point	^: up triangle
<code>axis([minx maxx miny maxy])</code>			
<code>xlabel, ylabel, title</code>	g: green	o: circle	<: left triangle
<code>bar, grid, scatter, corrcoef</code>	r: red	x: x	>: right triangle
<b>Lines</b>	c: cyan	+: plus	p: pentagram
-: solid	m: magenta	*: star	h: hexagram
:: dotted	y: yellow	s: square	: vertical line
-.: dash dot	k: black	d: diamond	_: horizontal line
—: dashed	w: white	v: down triangle	

### Functions

<code>function out = funcname(in)</code> <code>%define for out</code> <code>end</code>	Basic user defined function, % for comments
<code>function [x,y,z] = fname (a,b)</code> <code>%print, plot, display</code> <code>end</code>	Calling with only 1 output var gives 1st value only
<code>function fname (x,y)</code> <code>%print, plot, display</code> <code>end</code>	Input args are not always needed (call <code>fname</code> or <code>fname()</code> )
<code>primary function header</code> <code>body with sub function call</code> <code>end</code> <code>sub function header</code> <code>body</code> <code>end</code>	Subfunction can only be called by name of the primary
<code>function outs = pers(x)</code> <code>persistent runsum</code> <code>if isempty y(r unsum)</code> <code>runsum = 0;</code> <code>end</code> <code>runsum =runsum*x;</code> <code>outsum =runsum;</code> <code>end</code>	Persistent variables



### Functions (cont)

```
function out = ffname(x)
persistent R
if isempty(R)
R=0;
end
out=R+x;
R=x;
end
```

Adds to previous

```
function [x,y] = coor(t heta)
x=1;
y=2;
end
```

Function stub

### Matrix Manipulation

reshape(mat,n,m)	re-dimensions columnwise
fliplr	flips matrix left to right
flipud	flips matrix upside down
rot90	rotates matrix counterclockwise 90°
repmat(mat,n,m)	replicates n*m copies of mat
repelem(mat,n,m)	replicates elements from mat n*m times
cumsum/cumprod	cumulative sum/product
min(vec)	minimum vector value
max(vec)	maximum vector value
sum(vec)	sum vector elements
prod(vec)	product vector elements
abs(vec)	absolute vector values



