NumPy for MATLAB users

Help

MATLAB/Octave	Python	Description
doc help -i % browse with Info	help()	Browse help interactively
help help *or* doc doc	help	Help on using help
help plot	help(plot) *or* ?plot	Help for a function
help splines *or* doc splines	help(pylab)	Help for a toolbox/library package
demo		Demonstration examples

Searching available documentation

MATLAB/Octave	Python	Description
lookfor plot		Search help files
help	help(); modules [Numeric]	List available packages
which plot	help(plot)	Locate functions

Using interactively

MATLAB/Octave	Python	Description
octave -q	ipython -pylab	Start session
TAB *or* M-?	TAB	Auto completion
foo(.m)	execfile('foo.py') *or* run foo.py	Run code from file
history	hist -n	Command history
diary on [..] diary off		Save command history

exit *or* quit	CTRL-D
	CTRL-Z # windows
	sys.exit()
		End session

Operators

MATLAB/Octave	Python	Description
help -		Help on operator syntax

Arithmetic operators

MATLAB/Octave	Python	Description
a=1; b=2;	a=1; b=1	Assignment; defining a number
a + b	a + b *or* add(a,b)	Addition
a - b	a - b *or* subtract(a,b)	Subtraction
a * b	a * b *or* multiply(a,b)	Multiplication
a / b	a / b *or* divide(a,b)	Division

a .^ b	a ** b
	power(a,b)
	pow(a,b)
		Power, $a^b$

rem(a,b)	a % b
	remainder(a,b)
	fmod(a,b)
		Remainder

a+=1	a+=b *or* add(a,b,a)	In place operation to save array creation overhead
factorial(a)		Factorial, $n!$

Relational operators

MATLAB/Octave	Python	Description
a == b	a == b *or* equal(a,b)	Equal
a < b	a < b *or* less(a,b)	Less than
a > b	a > b *or* greater(a,b)	Greater than
a <= b	a <= b *or* less_equal(a,b)	Less than or equal
a >= b	a >= b *or* greater_equal(a,b)	Greater than or equal
a ~= b	a != b *or* not_equal(a,b)	Not Equal

Logical operators

MATLAB/Octave	Python	Description
a && b	a and b	Short-circuit logical AND
a || b	a or b	Short-circuit logical OR
a & b *or* and(a,b)	logical_and(a,b) *or* a and b	Element-wise logical AND
a | b *or* or(a,b)	logical_or(a,b) *or* a or b	Element-wise logical OR
xor(a, b)	logical_xor(a,b)	Logical EXCLUSIVE OR
~a *or* not(a) ~a *or* !a	logical_not(a) *or* not a	Logical NOT
any(a)		True if any element is nonzero
all(a)		True if all elements are nonzero

root and logarithm

MATLAB/Octave	Python	Description
sqrt(a)	math.sqrt(a)	Square root
log(a)	math.log(a)	Logarithm, base $e$ (natural)
log10(a)	math.log10(a)	Logarithm, base 10
log2(a)	math.log(a, 2)	Logarithm, base 2 (binary)
exp(a)	math.exp(a)	Exponential function

Round off

MATLAB/Octave	Python	Description
round(a)	around(a) *or* math.round(a)	Round
ceil(a)	ceil(a)	Round up

`floor(a)`	`floor(a)`	Round down
`fix(a)`	`fix(a)`	Round towards zero

Mathematical constants

MATLAB/Octave	Python	Description
`pi`	`math.pi`	$\pi=3.141592$
`exp(1)`	`math.e` *or* `math.exp(1)`	$e=2.718281$

Missing values; IEEE-754 floating point status flags

MATLAB/Octave	Python	Description
`NaN`	`nan`	Not a Number
`Inf`	`inf`	Infinity, $\infty$
	`plus_inf`	Infinity, $+\infty$
	`minus_inf`	Infinity, $-\infty$
	`plus_zero`	Plus zero, $+0$
	`minus_zero`	Minus zero, $-0$

Complex numbers

MATLAB/Octave	Python	Description
`i`	`z = 1j`	Imaginary unit
`z = 3+4i`	`z = 3+4j` *or* `z = complex(3,4)`	A complex number, $3+4i$
`abs(z)`	`abs(3+4j)`	Absolute value (modulus)
`real(z)`	`z.real`	Real part
`imag(z)`	`z.imag`	Imaginary part
`arg(z)`		Argument
`conj(z)`	`z.conj(); z.conjugate()`	Complex conjugate

Trigonometry

MATLAB/Octave	Python	Description
`atan(a,b)`	`atan2(b,a)`	Arctangent, $\arctan(b/a)$
	`hypot(x,y)`	Hypotenus; Euclidean distance

Generate random numbers

MATLAB/Octave	Python	Description

`rand(1,10)`	`random.random((10,))`
	`random.uniform((10,))`
		Uniform distribution

`2+5*rand(1,10)`	`random.uniform(2,7,(10,))`	Uniform: Numbers between 2 and 7
`rand(6)`	`random.uniform(0,1,(6,6))`	Uniform: 6,6 array
`randn(1,10)`	`random.standard_normal((10,))`	Normal distribution

Vectors

MATLAB/Octave	Python	Description
`a=[2 3 4 5];`	`a=array([2,3,4,5])`	Row vector, $1 \times n$-matrix

`adash=[2 3 4 5]';`	`array([2,3,4,5])[:,NewAxis]`
	`array([2,3,4,5]).reshape(-1,1)`
	`r_[1:10,'c']`
		Column vector, $m \times 1$-matrix

Sequences

MATLAB/Octave	Python	Description

`1:10`	`arange(1,11, dtype=Float)`
	`range(1,11)`
		1,2,3, ... ,10

`0:9`	`arange(10.)`	0.0,1.0,2.0, ... ,9.0
`1:3:10`	`arange(1,11,3)`	1,4,7,10
`10:-1:1`	`arange(10,0,-1)`	10,9,8, ... ,1
`10:-3:1`	`arange(10,0,-3)`	10,7,4,1
`linspace(1,10,7)`	`linspace(1,10,7)`	Linearly spaced vector of n=7 points
`reverse(a)`	`a[::-1]` *or*	Reverse
`a(:) = 3`	`a.fill(3), a[:] = 3`	Set all values to same scalar value

Concatenation (vectors)

MATLAB/Octave	Python	Description
`[a a]`	`concatenate((a,a))`	Concatenate two vectors
`[1:4 a]`	`concatenate((range(1,5),a), axis=1)`	

Repeating

MATLAB/Octave	Python	Description
`[a a]`	`concatenate((a,a))`	1 2 3, 1 2 3
	`a.repeat(3)` *or*	1 1 1, 2 2 2, 3 3 3
	`a.repeat(a)` *or*	1, 2 2, 3 3 3

Miss those elements out

MATLAB/Octave	Python	Description
`a(2:end)`	`a[1:]`	miss the first element
`a([1:9])`		miss the tenth element
`a(end)`	`a[-1]`	last element
`a(end-1:end)`	`a[-2:]`	last two elements

Maximum and minimum

MATLAB/Octave	Python	Description
`max(a,b)`	`maximum(a,b)`	pairwise max
`max([a b])`	`concatenate((a,b)).max()`	max of all values in two vectors
`[v,i] = max(a)`	`v,i = a.max(0),a.argmax(0)`	

Vector multiplication

MATLAB/Octave	Python	Description
`a.*a`	`a*a`	Multiply two vectors
`dot(u,v)`	`dot(u,v)`	Vector dot product, $u \cdot v$

Matrices

MATLAB/Octave	Python	Description
`a = [2 3;4 5]`	`a = array([[2,3],[4,5]])`	Define a matrix

Concatenation (matrices); rbind and cbind

MATLAB/Octave	Python	Description

`[a ; b]`	`concatenate((a,b), axis=0)`
	`vstack((a,b))`
		Bind rows

`[a , b]`	`concatenate((a,b), axis=1)`
	`hstack((a,b))`
		Bind columns

	`concatenate((a,b), axis=2)`
	`dstack((a,b))`
		Bind slices (three-way arrays)

`[a(:), b(:)]`	`concatenate((a,b), axis=None)`	Concatenate matrices into one vector

`[1:4 ; 1:4]`	`concatenate((r_[1:5],r_[1:5])).reshape(2,-1)`
	`vstack((r_[1:5],r_[1:5]))`
		Bind rows (from vectors)

`[1:4 ; 1:4]'`		Bind columns (from vectors)

Array creation

MATLAB/Octave	Python	Description
`zeros(3,5)`	`zeros((3,5),Float)`	0 filled array
	`zeros((3,5))`	0 filled array of integers
`ones(3,5)`	`ones((3,5),Float)`	1 filled array
`ones(3,5)*9`		Any number filled array
`eye(3)`	`identity(3)`	Identity matrix
`diag([4 5 6])`	`diag((4,5,6))`	Diagonal
`magic(3)`		Magic squares; Lo Shu
	`a = empty((3,3))`	Empty array

Reshape and flatten matrices

MATLAB/Octave	Python	Description

`reshape(1:6,3,2)';`	`arange(1,7).reshape(2,-1)`
	`a.setshape(2,3)`
		Reshaping (rows first)

`reshape(1:6,2,3);`	`arange(1,7).reshape(-1,2).transpose()`	Reshaping (columns first)
`a'(:)`	`a.flatten()` *or*	Flatten to vector (by rows, like comics)
`a(:)`	`a.flatten(1)`	Flatten to vector (by columns)
`vech(a)`		Flatten upper triangle (by columns)

Shared data (slicing)

MATLAB/Octave	Python	Description
`b = a`	`b = a.copy()`	Copy of a

Indexing and accessing elements (Python: slicing)

MATLAB/Octave	Python	Description

`a = [ 11 12 13 14 ...`
`21 22 23 24 ...`
`31 32 33 34 ]`
	`a = array([[ 11, 12, 13, 14 ],`
	`[ 21, 22, 23, 24 ],`
	`[ 31, 32, 33, 34 ]])`
		Input is a 3,4 array

`a(2,3)`	`a[1,2]`	Element 2,3 (row,col)
`a(1,:)`	`a[0,]`	First row
`a(:,1)`	`a[:,0]`	First column
`a([1 3],[1 4]);`	`a.take([0,2]).take([0,3], axis=1)`	Array as indices
`a(2:end,:)`	`a[1:,]`	All, except first row
`a(end-1:end,:)`	`a[-2:,]`	Last two rows
`a(1:2:end,:)`	`a[::2,:]`	Strides: Every other row
	`a[...,2]`	Third in last dimension (axis)
`a(:,[1 3 4])`	`a.take([0,2,3],axis=1)`	Remove one column
	`a.diagonal(offset=0)`	Diagonal

Assignment

MATLAB/Octave	Python	Description
`a(:,1) = 99`	`a[:,0] = 99`	
`a(:,1) = [99 98 97]'`	`a[:,0] = array([99,98,97])`	

`a(a>90) = 90;`	`(a>90).choose(a,90)`
	`a.clip(min=None, max=90)`
		Clipping: Replace all elements over 90

	`a.clip(min=2, max=5)`	Clip upper and lower values

Transpose and inverse

MATLAB/Octave	Python	Description
`a'`	`a.conj().transpose()`	Transpose
`a.'` *or* `transpose(a)`	`a.transpose()`	Non-conjugate transpose
`det(a)`	`linalg.det(a)` *or*	Determinant
`inv(a)`	`linalg.inv(a)` *or*	Inverse
`pinv(a)`	`linalg.pinv(a)`	Pseudo-inverse
`norm(a)`	`norm(a)`	Norms
`eig(a)`	`linalg.eig(a)[0]`	Eigenvalues
`svd(a)`	`linalg.svd(a)`	Singular values

`chol(a)`	`linalg.cholesky(a)`	Cholesky factorization
`[v,l] = eig(a)`	`linalg.eig(a)[1]`	Eigenvectors
`rank(a)`	`rank(a)`	Rank

Sum

MATLAB/Octave	Python	Description
`sum(a)`	`a.sum(axis=0)`	Sum of each column
`sum(a')`	`a.sum(axis=1)`	Sum of each row
`sum(sum(a))`	`a.sum()`	Sum of all elements
	`a.trace(offset=0)`	Sum along diagonal
`cumsum(a)`	`a.cumsum(axis=0)`	Cumulative sum (columns)

Sorting

MATLAB/Octave	Python	Description
`a = [ 4 3 2 ; 2 8 6 ; 1 4 7 ]`	`a = array([[4,3,2],[2,8,6],[1,4,7]])`	Example data
`sort(a(:))`	`a.ravel().sort()` *or*	Flat and sorted
`sort(a)`	`a.sort(axis=0)` *or* `msort(a)`	Sort each column
`sort(a')'`	`a.sort(axis=1)`	Sort each row
`sortrows(a,1)`	`a[a[:,0].argsort(),]`	Sort rows (by first row)
	`a.ravel().argsort()`	Sort, return indices
	`a.argsort(axis=0)`	Sort each column, return indices
	`a.argsort(axis=1)`	Sort each row, return indices

Maximum and minimum

MATLAB/Octave	Python	Description
`max(a)`	`a.max(0)` *or* `amax(a [,axis=0])`	max in each column
`max(a')`	`a.max(1)` *or* `amax(a, axis=1)`	max in each row
`max(max(a))`	`a.max()` *or*	max in array
`[v i] = max(a)`		return indices, i
`max(b,c)`	`maximum(b,c)`	pairwise max
`cummax(a)`		
	`a.ptp(); a.ptp(0)`	max-to-min range

Matrix manipulation

MATLAB/Octave	Python	Description
`fliplr(a)`	`fliplr(a)` *or* `a[:,::-1]`	Flip left-right
`flipud(a)`	`flipud(a)` *or* `a[::-1,]`	Flip up-down
`rot90(a)`	`rot90(a)`	Rotate 90 degrees

`repmat(a,2,3)`
`kron(ones(2,3),a)`
	`kron(ones((2,3)),a)`	Repeat matrix: [a a a ; a a a]

`triu(a)`	`triu(a)`	Triangular, upper
`tril(a)`	`tril(a)`	Triangular, lower

Equivalents to "size"

MATLAB/Octave	Python	Description
`size(a)`	`a.shape` *or* `a.getshape()`	Matrix dimensions
`size(a,2)` *or* `length(a)`	`a.shape[1]` *or* `size(a, axis=1)`	Number of columns
`length(a(:))`	`a.size` *or* `size(a[, axis=None])`	Number of elements
`ndims(a)`	`a.ndim`	Number of dimensions
	`a.nbytes`	Number of bytes used in memory

Matrix- and elementwise- multiplication

MATLAB/Octave	Python	Description
`a .* b`	`a * b` *or* `multiply(a,b)`	Elementwise operations
`a * b`	`matrixmultiply(a,b)`	Matrix product (dot product)
	`inner(a,b)` *or*	Inner matrix vector multiplication $a\cdot b'$
	`outer(a,b)` *or*	Outer product
`kron(a,b)`	`kron(a,b)`	Kronecker product
`a / b`		Matrix division, $b{\cdot}a^{-1}$
`a \ b`	`linalg.solve(a,b)`	Left matrix division, $b^{-1}{\cdot}a$ \newline (solve linear equations)
	`vdot(a,b)`	Vector dot product
	`cross(a,b)`	Cross product

Find; conditional indexing

MATLAB/Octave	Python	Description
`find(a)`	`a.ravel().nonzero()`	Non-zero elements, indices

`[i j] = find(a)`	`(i,j) = a.nonzero()`
	`(i,j) = where(a!=0)`
		Non-zero elements, array indices

`[i j v] = find(a)`	`v = a.compress((a!=0).flat)`
	`v = extract(a!=0,a)`
		Vector of non-zero values

`find(a>5.5)`	`(a>5.5).nonzero()`	Condition, indices
	`a.compress((a>5.5).flat)`	Return values
`a .* (a>5.5)`	`where(a>5.5,0,a)` *or* `a * (a>5.5)`	Zero out elements above 5.5
	`a.put(2,indices)`	Replace values

Multi-way arrays

MATLAB/Octave	Python	Description
`a = cat(3, [1 2; 1 2],[3 4; 3 4]);`	`a = array([[[1,2],[1,2]], [[3,4],[3,4]]])`	Define a 3-way array
`a(1,:,:)`	`a[0,...]`	

File input and output

MATLAB/Octave	Python	Description

`f = load('data.txt')`	`f = fromfile("data.txt")`
	`f = load("data.txt")`
		Reading from a file (2d)

`f = load('data.txt')`	`f = load("data.txt")`	Reading from a file (2d)
`x = dlmread('data.csv', ';')`	`f = load('data.csv', delimiter=';')`	Reading fram a CSV file (2d)
`save -ascii data.txt f`	`save('data.csv', f, fmt='%.6f', delimiter=';')`	Writing to a file (2d)
	`f.tofile(file='data.csv', format='%.6f', sep=';')`	Writing to a file (1d)
	`f = fromfile(file='data.csv', sep=';')`	Reading from a file (1d)

Plotting

Basic x-y plots

MATLAB/Octave	Python	Description
`plot(a)`	`plot(a)`	1d line plot
`plot(x(:,1),x(:,2),'o')`	`plot(x[:,0],x[:,1],'o')`	2d scatter plot
`plot(x1,y1, x2,y2)`	`plot(x1,y1,'bo', x2,y2,'go')`	Two graphs in one plot

`plot(x1,y1)`
`hold on`
`plot(x2,y2)`
	`plot(x1,y1,'o')`
	`plot(x2,y2,'o')`
	`show() # as normal`
		Overplotting: Add new plots to current

`subplot(211)`	`subplot(211)`	subplots
`plot(x,y,'ro-')`	`plot(x,y,'ro-')`	Plotting symbols and color

Axes and titles

MATLAB/Octave	Python	Description
`grid on`	`grid()`	Turn on grid lines

`axis equal`
`axis('equal')`
`replot`
	`figure(figsize=(6,6))`	1:1 aspect ratio

`axis([ 0 10 0 5 ])`	`axis([ 0, 10, 0, 5 ])`	Set axes manually

`title('title')`
`xlabel('x-axis')`
`ylabel('y-axis')`
		Axis labels and titles

	`text(2,25,'hello')`	Insert text

Log plots

MATLAB/Octave	Python	Description
`semilogy(a)`	`semilogy(a)`	logarithmic y-axis
`semilogx(a)`	`semilogx(a)`	logarithmic x-axis
`loglog(a)`	`loglog(a)`	logarithmic x and y axes

Filled plots and bar plots

MATLAB/Octave	Python	Description

`fill(t,s,'b', t,c,'g')`
`% fill has a bug?`
	`fill(t,s,'b', t,c,'g', alpha=0.2)`	Filled plot

Functions

MATLAB/Octave	Python	Description
`f = inline('sin(x/3) - cos(x/5)')`		Defining functions

`ezplot(f,[0,40])`
`fplot('sin(x/3) - cos(x/5)', [0,40])`
`% no ezplot`
	`x = arrayrange(0,40,.5)`
	`y = sin(x/3) - cos(x/5)`
	`plot(x,y, 'o')`
		Plot a function for given range

Polar plots

MATLAB/Octave	Python	Description

`theta = 0:.001:2*pi;`
`r = sin(2*theta);`
	`theta = arange(0,2*pi,0.001)`
	`r = sin(2*theta)`	

`polar(theta, rho)`	`polar(theta, rho)`	

Histogram plots

MATLAB/Octave	Python	Description
`hist(randn(1000,1))`		
`hist(randn(1000,1), -4:4)`		
`plot(sort(a))`		

3d data

Contour and image plots

MATLAB/Octave	Python	Description

`contour(z)`	`levels, colls = contour(Z, V, origin='lower', extent=(-3,3,-3,3))`
	`clabel(colls, levels, inline=1, fmt='%1.1f', fontsize=10)`
		Contour plot

`contourf(z); colormap(gray)`	`contourf(Z, V, cmap=cm.gray, origin='lower', extent=(-3,3,-3,3))`	Filled contour plot

`image(z)`
`colormap(gray)`
	`im = imshow(Z, interpolation='bilinear', origin='lower', extent=(-3,3,-3,3))`	Plot image data

	`# imshow() and contour() as above`	Image with contours

`quiver()`	`quiver()`	Direction field vectors

Perspective plots of surfaces over the x-y plane

MATLAB/Octave	Python	Description

`n=-2:.1:2;`
`[x,y] = meshgrid(n,n);`
`z=x.*exp(-x.^2-y.^2);`
	`n=arrayrange(-2,2,.1)`
	`[x,y] = meshgrid(n,n)`
	`z = x*power(math.e,-x**2-y**2)`	

`mesh(z)`		Mesh plot

`surf(x,y,z)` *or* `surfl(x,y,z)`
`% no surfl()`
		Surface plot

Scatter (cloud) plots

MATLAB/Octave	Python	Description
`plot3(x,y,z,'k+')`		3d scatter plot

Save plot to a graphics file

MATLAB/Octave	Python	Description

`plot(1:10)`
`print -depsc2 foo.eps`
`gset output "foo.eps"`
`gset terminal postscript eps`
`plot(1:10)`
	`savefig('foo.eps')`	PostScript

	`savefig('foo.pdf')`	PDF
	`savefig('foo.svg')`	SVG (vector graphics for www)
`print -dpng foo.png`	`savefig('foo.png')`	PNG (raster graphics)

Data analysis

Set membership operators

MATLAB/Octave	Python	Description

`a = [ 1 2 2 5 2 ];`
`b = [ 2 3 4 ];`
	`a = array([1,2,2,5,2])`
	`b = array([2,3,4])`
	`a = set([1,2,2,5,2])`
	`b = set([2,3,4])`
		Create sets

`unique(a)`	`unique1d(a)`
	`unique(a)`
	`set(a)`
		Set unique

`union(a,b)`	`union1d(a,b)`
	`a.union(b)`
		Set union

`intersect(a,b)`	`intersect1d(a)`
	`a.intersection(b)`
		Set intersection

`setdiff(a,b)`	`setdiff1d(a,b)`
	`a.difference(b)`
		Set difference

`setxor(a,b)`	`setxor1d(a,b)`
	`a.symmetric_difference(b)`
		Set exclusion

`ismember(2,a)`	`2 in a`
	`setmember1d(2,a)`
	`contains(a,2)`
		True for set member

Statistics

MATLAB/Octave	Python	Description

`mean(a)`	`a.mean(axis=0)`
	`mean(a [,axis=0])`
		Average

`median(a)`	`median(a)` *or* `median(a [,axis=0])`	Median
`std(a)`	`a.std(axis=0)` *or* `std(a [,axis=0])`	Standard deviation
`var(a)`	`a.var(axis=0)` *or* `var(a)`	Variance
`corr(x,y)`	`correlate(x,y)` *or* `corrcoef(x,y)`	Correlation coefficient
`cov(x,y)`	`cov(x,y)`	Covariance

Interpolation and regression

MATLAB/Octave	Python	Description

`z = polyval(polyfit(x,y,1),x)`
`plot(x,y,'o', x,z ,'-')`
	`(a,b) = polyfit(x,y,1)`
	`plot(x,y,'o', x,a*x+b,'-')`
		Straight line fit

`a = x\y`	`linalg.lstsq(x,y)`	Linear least squares $y = ax + b$
`polyfit(x,y,3)`	`polyfit(x,y,3)`	Polynomial fit

Non-linear methods

Polynomials, root finding

MATLAB/Octave	Python	Description
	`poly()`	Polynomial
`roots([1 -1 -1])`	`roots()`	Find zeros of polynomial

`f = inline('1/x - (x-1)')`
`fzero(f,1)`
		Find a zero near $x = 1$

`solve('1/x = x-1')`		Solve symbolic equations
`polyval([1 2 1 2],1:10)`	`polyval(array([1,2,1,2]),arange(1,11))`	Evaluate polynomial

Differential equations

MATLAB/Octave	Python	Description
`diff(a)`	`diff(x, n=1, axis=0)`	Discrete difference function and approximate derivative
		Solve differential equations

Fourier analysis

MATLAB/Octave	Python	Description
`fft(a)`	`fft(a)` *or*	Fast fourier transform

`ifft(a)`	`ifft(a)` *or* `convolve(x,y)`	Inverse fourier transform Linear convolution

Symbolic algebra; calculus

MATLAB/Octave	**Python**	**Description**
`factor()`		Factorization

Programming

MATLAB/Octave	**Python**	**Description**
`.m`	`.py`	Script file extension

`%`
`%` *or* `#`
	`#`	Comment symbol (rest of line)

`% must be in MATLABPATH`
`% must be in LOADPATH`
	`from pylab import *`	Import library functions

`string='a=234';`
`eval(string)`
	`string="a=234"`
	`eval(string)`
		Eval

Loops

MATLAB/Octave	**Python**	**Description**
`for i=1:5; disp(i); end`	`for i in range(1,6): print(i)`	for-statement

`for i=1:5`
`disp(i)`
`disp(i*2)`
`end`
	`for i in range(1,6):`
	`print(i)`
	`print(i*2)`
		Multiline for statements

Conditionals

MATLAB/Octave	**Python**	**Description**
`if 1>0 a=100; end`	`if 1>0: a=100`	if-statement
`if 1>0 a=100; else a=0; end`		if-else-statement

Debugging

MATLAB/Octave	**Python**	**Description**
`ans`		Most recent evaluated expression
`whos` *or* `who`		List variables loaded into memory
`clear x` *or* `clear [all]`		Clear variable $x$ from memory
`disp(a)`	`print a`	Print

Working directory and OS

MATLAB/Octave	**Python**	**Description**
`dir` *or* `ls`	`os.listdir(".")`	List files in directory
`what`	`grep.grep("*.py")`	List script files in directory
`pwd`	`os.getcwd()`	Displays the current working directory
`cd foo`	`os.chdir('foo')`	Change working directory
`!notepad`	`os.system('notepad')`	Invoke a System Command
`system("notepad")`	`os.popen('notepad')`	

Time-stamp: "2007-11-09T16:46:36 vidar"