

```
%Matlab/Octave tutorial

%Adapted from Mike Kahana

%Go ahead and type this in line-by-line and just see what
%happens! (Some commands in this tutorial depend on previous
%commands, so do it all in one session.)

%The percent sign is a comment and the rest of the line does
%nothing.
%help gives documentation
help help
help +

%data structures
%scalar
3

%vector
[3 4 5]
%or
[3
4
5]

%matrix
[1 2 3
4 5 6
7 8 9]

%basic operators and syntax
%assignment
scalar = 5
vect = [3 4 5]
mat = [1 2 3
4 5 6
7 8 9]
% or
mat = [1 2 3; 4 5 6; 7 8 9]
% etc.

%use ; to end a line and so that Octave does not output to the
%screen
mat = [1 2 3; 4 5 6; 7 8 9];

%indexing into vectors and matrices
%vector(index), matrix(row, column)
vect(2)
mat(2, 3)

%colon means entire row or column
%so mat(2, :) = [4 5 6]
```

```

mat(2, :)
mat(:, 2)

%or a:b = from a to b
%1:5 = [1 2 3 4 5]
%vect(1:2) = [3 4]
1:5
vect(1:2)

%move by a step other than 1 by using 2 colons
10:2:20 %from 10 to 20 by 2s

%end means the last element of a row or column
%so vect(2:end) = [4 5];
vect(2:end)

%an apostrophe ' trasposes a vector or matrix
%vect' = [3
% 4
% 5]
vect'

%concatenation
x = [1 2 3]
y = [4 5 6]

[x y] %horizontal
[x; y] %vertical

%basic operators, +, -, *, /
%on scalars these work as expected
3 + 4
2 * 2
scalar + scalar

% ^ is the power operator
2^3

%on vectors and matrices they use linear algebra
x = [1 2 3]
y = [4 5 6]
%dot product, if you don't know what a dot product is, look it
up! It is very useful!
x*y'
%cross product, same, look it up!
x'*y

%if you want to mutliply and divide, element by element use dot
functions, .* ./ .^
x.*y

%logical operators

```

```

% >, <, ==, <=, >=, | (or), & (and), ~ (not)
%these work on vectors and matrices as well as scalars
%.2 < .3 will return 1
%.3 < .2 will return 0

% logical AND 0&0=0, 0&1=0, 1&0=0, 1&1=1
1&1
0&1

% logical OR 0|0=0, 0|1=1, 1|0=1, 1|1=1
0|0
0|1

%rand gives a random number between 0 and 1.
rand
%rand(2, 3) gives a 2x3 matrix of random numbers
rand(2, 3)

%rand < .3 will return 1 if a random number was less than .3
%rand(1, 10) < .3 will return on average 3 of 10 ones. The rest
%will be zeros.
rand < .3
rand(1, 10) < .3

%basic loops and conditionals
%if, else
x = 3
if(x==3)
disp('it is three')
elseif (x==4)
disp('it is four')
else
disp('it is not three or four')
end
% note: == not =,
% = assignment as in scalar = 3
% == test as in x == 3
% BE VERY CAREFUL ABOUT THIS

%while
x = 1
while x < 5
disp(x);
x = x+1;
end

%for
for x=1:5
disp(x);
end

for x=1:.1:2

```

```
disp(x);
end

%useful math functions

%mean of a vector or means of each row or column of a matrix
mean(vect)

%standard deviation of a vector or of each row or column of a
%matrix
std(vect)

%uniformly distributed random numbers between zero and 1
rand
rand(1, 2)

%normally distributed random numbers between zero and 1
randn
randn(3, 4)

%absolute value
abs(-3)
abs([-3 2 0])

%minimum
min(vect)

%maximum
max(vect)

%square root
sqrt(4)

%natural log
log(10)

%modulus (remainder)
mod(10,3)

%e^ for example, to raise e to the second power call exp(2)
exp(2)

%summation
sum(vect)

%useful variables
pi %pi
NaN %not a number ex: 0/0 = NaN
Inf %Infinity ex: 5/0 = Inf
ans %the last octave answer, use as a variable

%other useful functions
```

```
%display a variable or a string
disp('hello')
disp(vect)

%largest dimension of a vector or matrix
length(vect)

%vector of the lengths of each dimension
size(mat)

%vector or matrix of zeros (you specify the size)
zeros(2, 3)

%vector or matrix of ones (you specify the size)
ones(3, 2)

%check whether a vector or matrix is empty []
isempty(vect)
isempty([])

%sorts a vector
sort(vect)

%copies (repeats) a scalar, vector, or matrix for a given number
of rows and columns
repmat(vect, 2, 3)

%lists defined symbols
who

%lists defined symbols and their size
whos

%clears defined symbols
clear scalar
scalar
scalar = 5
scalar

%saves all of your variables to a file
save tmp % save
clear % clear variables
scalar % see, they are gone

%loads the variables from a file.
load tmp % load them
scalar % they are back

%to find out where your files are saved (find the working
%directory.
pwd
```

```
%To change the working directory use 'cd <new directory name>'
% . is the current directory, .. is one directory up

%find finds the nonzero indices of a vector or matrix
%examples:
x = [6 7 8 9 10]
y = [2 3 5 7 11]
find(x>8) %[4 5]
x(find(x>8)) %[9 10]
x(find(x>7 & y<11)) %[8 9]
y(find(x>7 & y<11)) %[5 7]

%up-arrow scrolls through command list

%plotting

%line plots
x = [-5:5]
y = x.^2
plot(x, y)

%open a new plotting window
figure

%you can plot multiple graphs in the same window by typing
%'hold on'
%to make sure that any new plots don't overlap with this one,
%you can type 'hold off'

%errorbars (Works slightly differently in octave and matlab (last
%I checked), see help errorbar)
M = rand(5, 10)
index = 1:10
avg = mean(M)
errors = std(M)
errorbar(index, avg, errors)

%histograms
x = randn(1, 100)
figure

hist(x)

%bar plots
x = [1 2 3 4]
y = [1 10 5 2]
figure

bar(x, y)

%surface plots
```

```

x = repmat(1:10, 10, 1)
y = repmat([1:10]', 1, 10)
z = (x.^2).*y
figure
mesh(x, y, z)

%scatter plots (not implemented on macs yet, might work on PCs)
x = 1:10
y = x.^2 + randn(1, 10)
figure
scatter(x, y)

%useful plotting functions
%labels the x axis
xlabel('tmp1')
%labels the y axis
ylabel('tmp2')
%gives the graph a title
title('tmp3')

%To set the x-range to -10 to 10 and y-range to 0 to 10
axis([-10 10 0 10])

%functions (and scripts)
%A function is a bit of code that is separate from other code and
%can be used over and over.
%In Octave, a function is an "m-file". That is, just a file with
%.m at the end. That tells Octave that it is a function.
%MAKE SURE YOU SAVE YOUR FILE AS TEXT ONLY WITH LINE BREAKS. If
%you save it in any other format, Octave may give you odd errors.
%Function syntax works as follows
%function returnVariable = nameofFunction(parameter1, parameter2,
%etc)

function ret = fib(N, show)
a = 0;
b = 1;
c = zeros(1, N);
for i=1:N
c(i) = a + b;
a = b;
b = c(i);
end
if show % Plots if show is not 0, otherwise doesn't plot
plot(1:N, c)
end
ret = c(end);

%You can save this as an m-file and run it as a function from the
%command line. This should be saved as a file, not typed in!

```

```

%Just put the function in a text editor and save with a .m
%extension. (You cannot just type functions in, you must save
%them to an m-file.) Then type
fib(5, 1)

%Octave needs to be able to find your function.
%To tell Octave where it is, you may need to put it in
%Octave's search path.

%Setting the path to display or change octave's search path
path

%for mac, to put my desktop directory into the path
addpath('/Users/andrewcohen/Desktop');

%for windows
addpath(':\Documents and Settings\default\desktop\');

% Hitting the up-arrow will get you previously typed commands.
% If stuck, CNTRL-. Or CMD-. should stop Octave from running.

%NASTY COMMENTS FROM MATLAB

1/0
Warning: Divide by zero.
% means: you are dividing by zero

x = [1 2 3 4]

x(5)
??? Index exceeds matrix dimensions.
% means: x does not have anything at 5

x(0)
??? Index into matrix is negative or zero.
% means: the first item in x is at position 1 and nothing can be
at zero

x(-1)
??? Index into matrix is negative or zero.
% means: a vector cannot have negative indices

x(.6)
Warning: Subscript indices must be integer values.
% means: MATLAB will round up your decimal index however it
prefers that you enter an integer

x = [1 2 3 4]
y = [1 2 3]

x.*y

```

```
??? Error using ==> .*  
Matrix dimensions must agree.  
% means: x and y are not the same size  
  
%quitting octave (all your variables will be lost  
%if they are not saved).  
exit
```