Quick MATLAB Syntax Guide

Some useful things, not everything

if-Statement Structure:

```
if (a<0)          % general structure of if-statement
    x = 1             %calculation when if true
end               % closing statement of if statement
```

logical comparators					
<	>	<=	>=	==	~=

if-elseif-else Statment Structure:

```
if (a<0)          %  general structure of if-statement
  x=1               % calculation for if true
 elseif(a>0)      %  elseif general statement
  x=2               % calculation for elseif true
else              %  else statement
  x=3               % calculation when no if or elseif is true
end               % closing statement of if statement
```

logical operators	
~	NOT
&	AND
|	OR

while Loop Structure:

```
while (k>0)       % enter loop when while statement is true
    k=k-1             % calculation when while is true
end               %  closing statement of while statement
```

for Loop Structure:

```
for k=1:0.001:10        %  enter for loop, where  k =first:increment:last
    x=x+5             %  calculation
    if (k>7)          %  option method to exit loop before the end of the for loop
      break               % send program to statement following end%for
    end             %closing statement of if
end               % closing statement of for
```

function Structure:

```
function [output1,output2] = practice(input1, input2)

output1 = input1 * 5;
output2 = input2 * 10;

end  % optional to end a function
```

Simple Graphics Commands:

plot(x,y)	Plots array y vs. array x
plot(x,y,'*b')	Plots array y vs. array x with blue *'s
plot(x,y,'*b',x,z,'-r')	Plots y vs. x and z vs. x on same graph
grid	Creates a grid background on graph
xlabel('x axis label')	Labels x axis
ylabel('y axis label')	Labels y axis
title('title name')	Puts title on plot
text(gx,gy,'string')	Places text 'string' at position (gx,gy) on graph
legend('string1','string2', ...)	Creates legend of multiple plot lines on graph
axis ([xmin xmax ymin ymax])	Manually sets axis limits
axis auto	Automatically sets axis limits (default)
axis square	Fixes graph in square shape
axis equal	Fixes graph to equal x and y increments
axis normal	Resets graph to default mode
whitebg('r')	Change graph background color to red
hold on	Freezes current graph for appending other plots to it.
hold off	Removes freeze to plot
shg	Shows graphic window
clg	Clears graphic window
close	Closes the graphic window

Operators and Other special characters:

=	Assignment
[ ]	designates matrix beginning and end
( )	parenthesis, brackets subscripts, brackets arguments of functions
' '	delimiters for character string
, or space	separates elements in row of an array
; or return	separates rows of an array
...	continues command on next line
%	designates following characters as comments
:	designate a range of values in row
*	Full matrix multiplication
.*	Array multiplication
'	Transpose of a matrix

Trigonometric functions: (Trig function use radians, not degrees)

cos(x)	cosine of x
sin(x)	sine of x
tan(x)	tangent of x

Common Math Functions:

abs(x)	absolute value of x
exp(x)	exponential operation, e^x
log(x)	natural logarithm
log10(x)	base 10 logarithm
rem(x,y)	computes remainder of division for x/y - mod(x,y) also works
sign(x)	generates -1, 0, or 1 based on sign of x
sqrt(x)	square root of x
ceil(x)	round x toward +infinity
floor(x)	round x toward -infinity
round(x)	round x to nearest integer

Utility matrices and matrix fucntions: (m,n are integers, v is vector, A is a matrix)

linspace(x,y,m)	linearly spaced vector from x to y with m points
magic(m)	returns an m*m magic square with integers from 1 to m^2
zeros(m,n)	returns an m*n matrix of zeros
ones(m,n)	returns an m*n matrix of ones
rand(m,n)	returns an m*n matrix of random numbers between 0 and 1
length(V)	returns length of vector V
size(A)	returns size of matrix A as vector [#of row, # of col]

Character String Functions:

where S is a string, n is an integer, x is a real number, A is a matrix

abs(S)	converts characters in string S to their ASCII numeric values
char(n)	converts ASCII numerical values (integer matix) to character arrays.
lower(S)	converts any upper case letters in the string S to lower case
upper(S)	converts lower case letters in the string S to upper case.
mat2stsr(A)	converts matrix A to a string
num2str(x)	converts number x to string characters
strcmp(S,T)	compares two strings S and T, returns 1 if same
strncmp(S,T,n)	compares first n characters in given strings S and T
strcat(S,T)	concatenates strings S and T horizontally, ignoring trailing blanks

Output to Screen Statements:

fprintf('%f \n',m)	prints values of matrix, m with line feed
fprintf('%i', n)	prints values of integer n (%d also works)
fprintf('%f %f %f \n',x,y,z)	print out values of x,y, and z on same line
fprintf('The string is %s \n',S)	print out string S in position %s.

Input from Keyboard

x=input('enter value for x: ')	prompts the user to input a value for variable x
S=input('enter a string','s')	prompts the user to input a string to variable S
k=menu('title','button1','button2','button3')	creates menu with 3 button, puts integer in k
menu('message','Continue')	creates a button which requires user to click to continue

Format field characters for fprint and fscanf commands:

Conversion Characters		Escape Characters	
%f	fixed point notation	\n	new line
%8.4f	fixed point with 8 space min, 4 dec. places	\t	horizontal tab
%-f	fixed point notation, left justified	\b	backspace
%e	exponential notional	\r	carriage return
%12.4e	exp. notation with 12 space min, 4 dec. places	\f	form feed
%g	either fixed point or exp. notation.	\\	backslash
%i	integer notation		
%3i	integer notation with 3 spaces min		
%s	character string notation		

File Access Commands:

fid = fopen('filename.ext','permission') Opens file with name filename.ext ,
fclose(fid) Close file.
frewind(fid) Reset file pointer to beginning of file.
A=fscanf(fid,'%f') Read floating point numbers from file storing values into column matrix A until all numbers have been read.
B=fscanf(fid,'%f', 6) Read 6 floating point numbers from file storing values into column matrix B
[A,count]=fscanf(fid,'%f',[2,inf]) Read floating point numbers from file storing them in 2xN matrix A until all values have been read.
fprintf(fid,'%f \n', B) Write all values of B matrix into the file, one item per line.
fprintf(fid,'%f %f %f \n', a,b,c) Write values a, b, and c into the file on same line.

permissions:
'w' write file
'r' read file
'a' append file

Example of using fscanf to read in values of x and y from data file: Bob.dat on A drive.

```
fidBob=fopen('A:Bob.dat','r')
A=fscanf(fidBob,'%f%f ',[2,5])
x = A(1,1:end)
y = A(2,1:end)
fclose(fidBob)
```

Bob.dat

1	5
2	10
3	15
4	25
5	20

this gives x = [1 2 3 4 5] and y = [5 10 15 25 20]

Example used to read long one dimensional array of unknown size in file: data.txt

```
fidA=fopen('data.txt','r')
[A,count]=fscanf(fidA,'%f')
fclose(fidA)
```

The data gets stored in the column vector A and the variable count is the number of data values read.

Example used to write the values of x and y arrays into a file called MyData.txt

```
x = [1 2 3 4 5]
y = [ 5 10 35 50 100]
fidMy=fopen('MyData.txt','w')
fprintf(fidMy,'MyData.txt starts here:')
for i=1:length(x)
    fprintf(fidMy,' %8.4f %f \n', x(i), y(i))
end %for
fclose(fidMy)
```

When completed the written file will look like

```
MyData.txt starts here:
  1.0000      5.0000
  2.0000     10.0000
  3.0000     35.0000
  4.0000     50.0000
  5.0000    100.0000
```