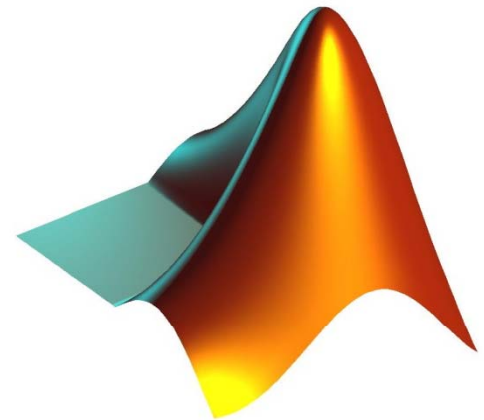




MATLAB

An Introduction





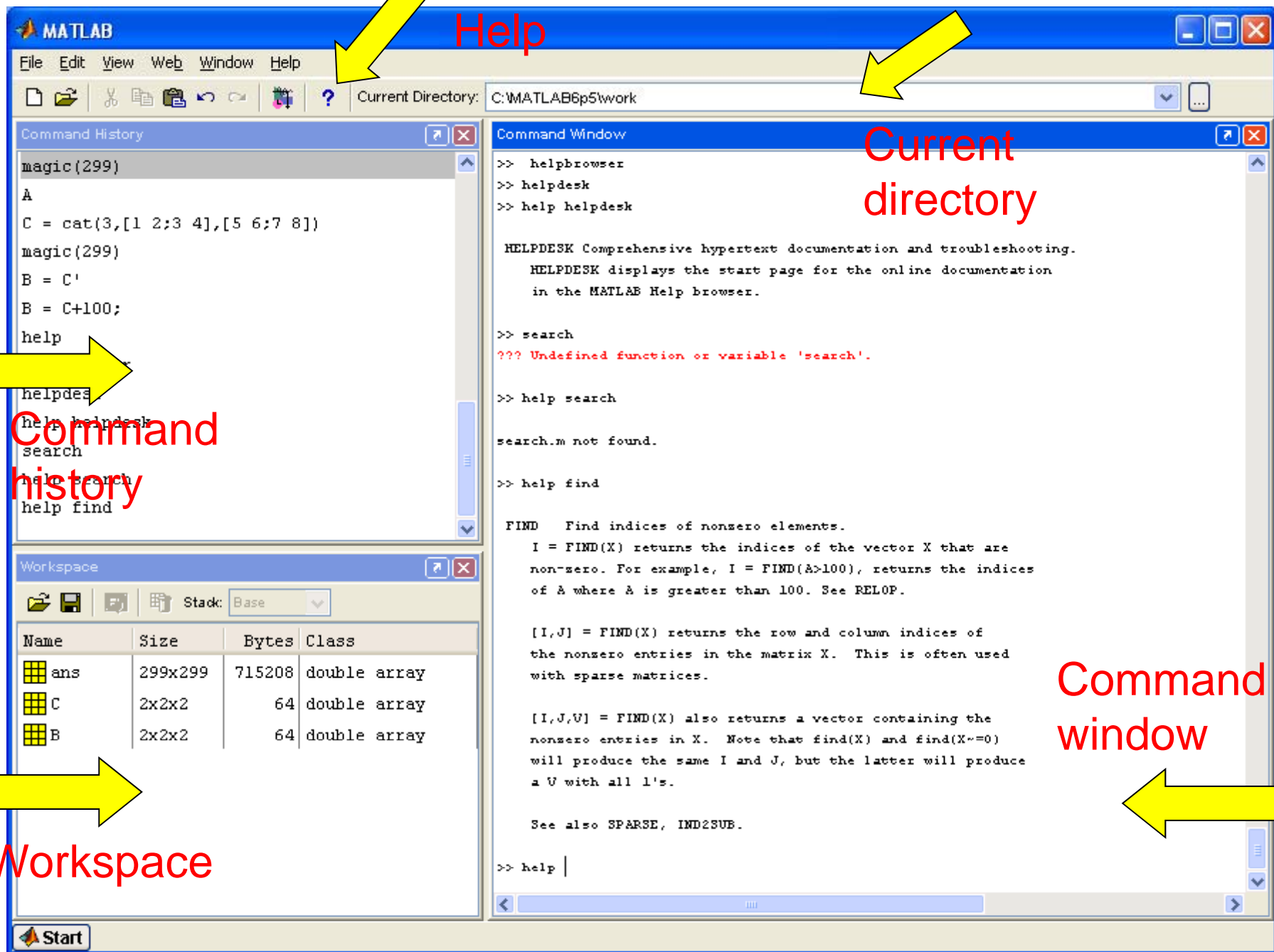
What is MATLAB?

- MATLAB stands for **MAT**rix **Lab**oratory.
- It was originally developed as a user interface to call some FORTRAN matrix routines.
- Now, MATLAB is far more than “MAT LAB”.
- MATLAB® is a high-performance language for **technical** computing. It integrates **computation**, **visualization**, and **programming**.
- Matlab program and script files always have filenames ending with “.m”.



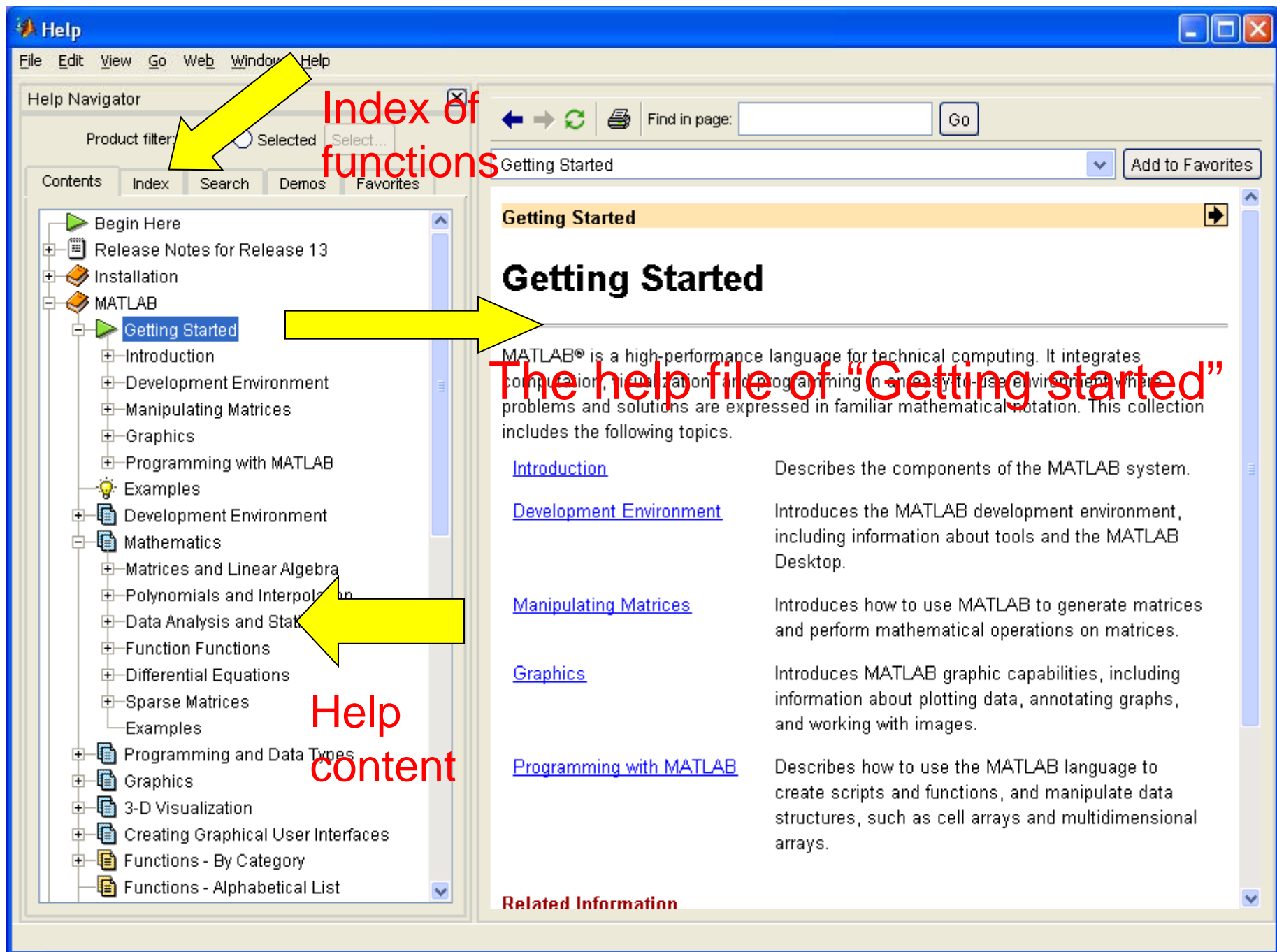
Matlab Tutorial Materials

- Type “Matlab tutorial” in Google search, you will find many websites.
- Click 2 or 3 of them and read to get some basic ideas of Matlab.
- The **best** way to learn Matlab is to **do more programming**.





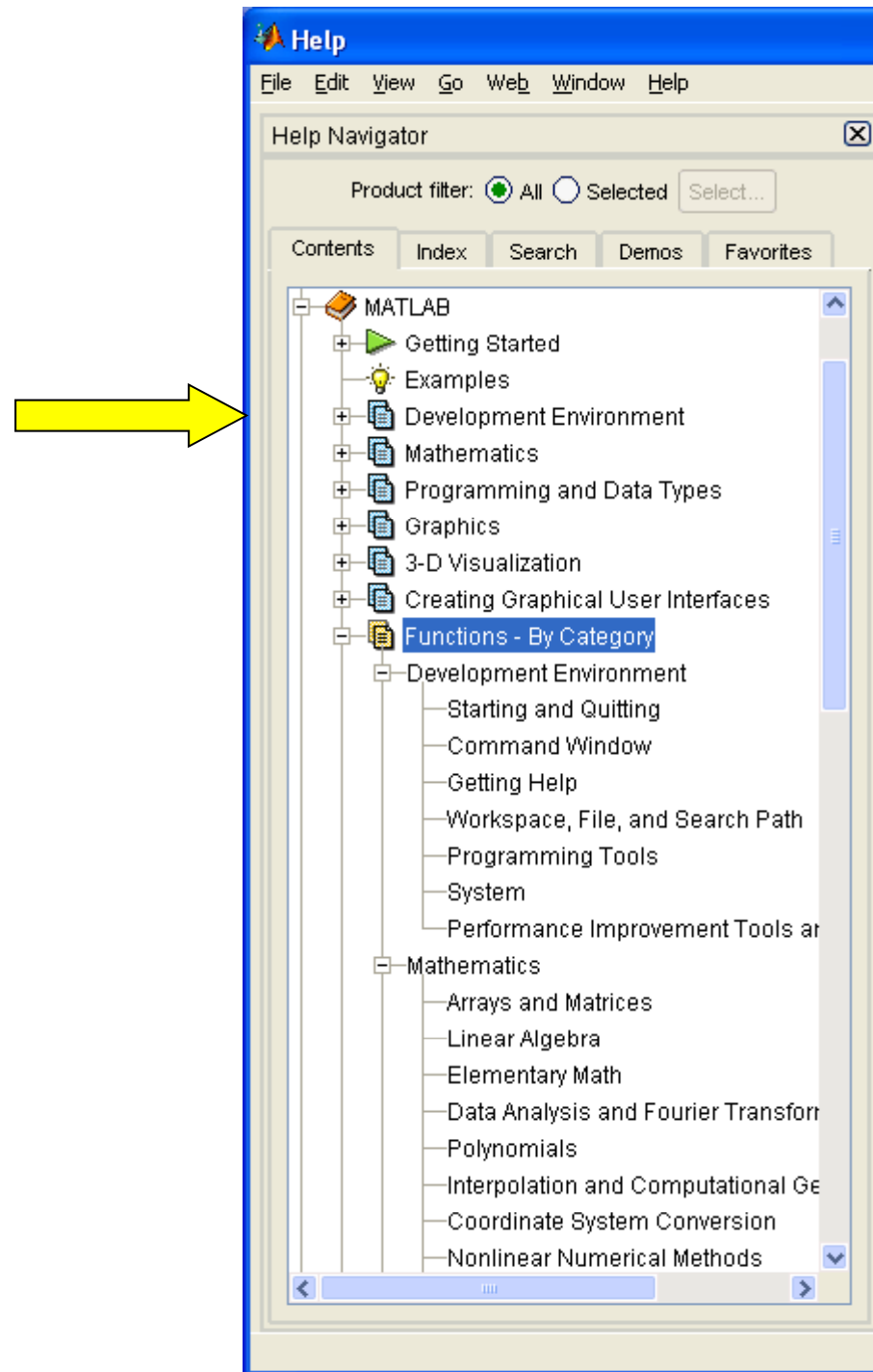
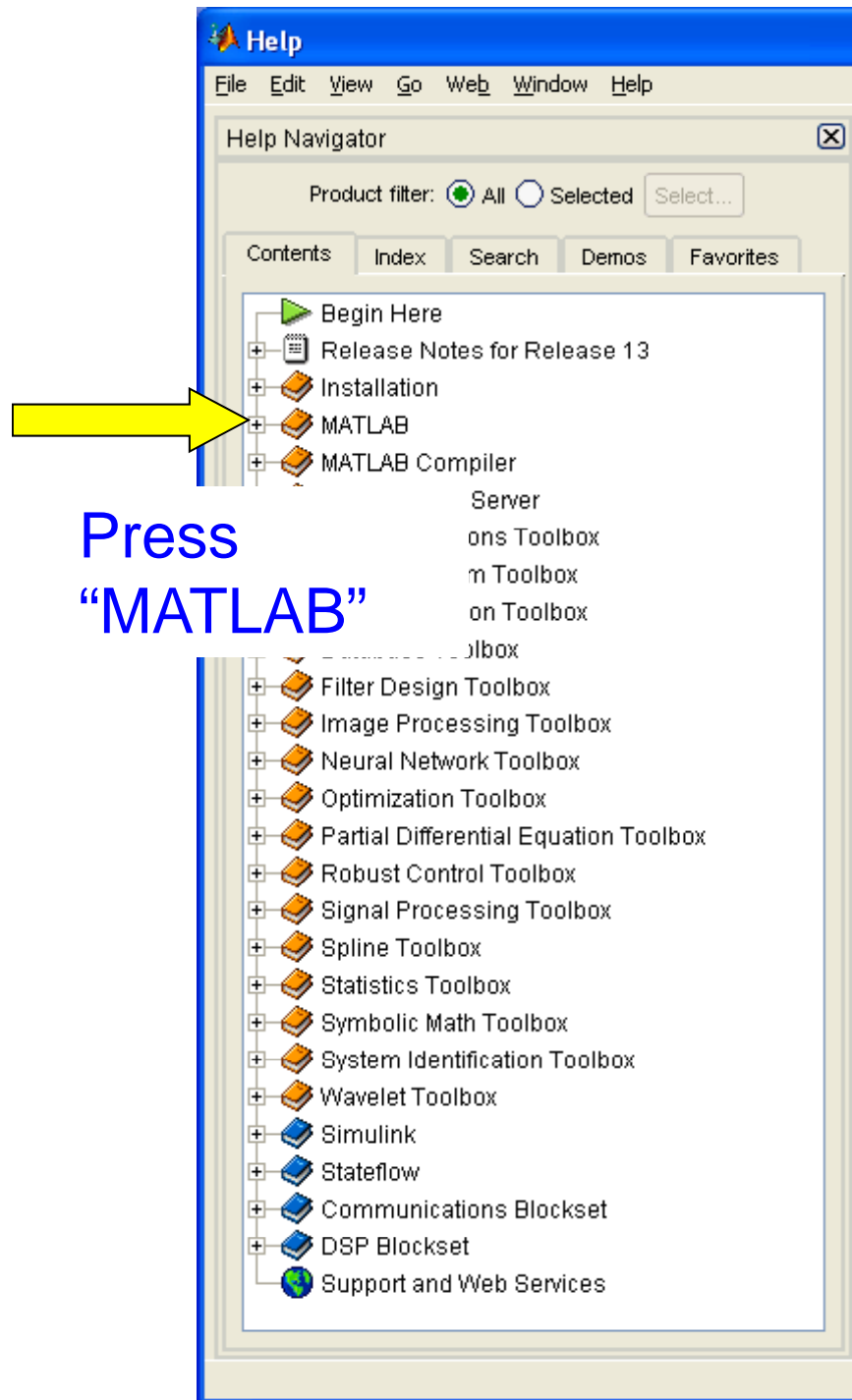
- **Workspace** is a tool to display all variables
- When start MATLAB, no variables is defined
- When close MATLAB, all variables are gone
- Related commands
 - **clear**: clear variables and functions from memory
 - **who**: list current variables.
 - **load** *filename*: load workspace variables from disk
 - **save** *filename*: save workspace variables to disk





Getting Help

- helpbrowser
- Use index, search in help browser
- Read the PDF document
- The Internet
<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>
- help command: help *filename*





Define a Matrix

- A matrix is a **two-dimensional** array of real or complex numbers
 - A number is a 1×1 matrix
 - A column vector is $n \times 1$ matrix
 - A row vector is a $1 \times n$ matrix



Matrix, vector and scalar:

A **matrix** is a collection of numerical values that are organized into a specific configuration of rows and columns.

Example:

```
A=[ 1 2 3 4  
    5 6 7 8];
```

A has 2 rows and 4 columns, i.e. it is a 2 x 4 matrix which has 8 elements in total.

A **scalar** is represented by a 1 x 1 matrix in **Matlab**: `a=1;`



Matrix, vector and scalar:

$x = [3.5, 33.22, 24.5] ;$ x is a **row** vector or 1×3 matrix

$x1 = \begin{bmatrix} 2 \\ 5 \\ 3 \\ -1 \end{bmatrix} ;$ $x1$ is **column** vector or 4×1 matrix

Matlab is "case sensitive", that is, it treats the name 'C' and 'c' as two different variables.



Colon operator

If two integer numbers are separated by a colon, **Matlab** will generate all of the integers between these two integers.

$a = 1:8$

generates row vector $a = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]$.

If three numbers, integer or non-integer, are separated by two colons, the **middle** number is interpreted to be a “**step**” and the **first and third** are interpreted to be “**limits**”:

$b = 0 : 0.2 : 1.0$

generates row vector $b = [0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1.0]$



The colon operator can be used to create a vector from a matrix.

$$x = \begin{bmatrix} 2 & 6 & 8 \\ 0 & 1 & 7 \\ -2 & 5 & -6 \end{bmatrix}$$

The command `y = x(:,1)` creates the column vector

$$y = \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} \quad \text{"(:,1)" means all rows and the first column}$$

The command `z = x(1,:)` creates the row vector

$$z = \begin{bmatrix} 2 & 6 & 8 \end{bmatrix} \quad \text{"(1,:)" means all columns and the first row}$$



The colon operator can also extract sub-matrices from larger matrices.

$$c = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix}$$

Then what is $d = c(:,2:3)$?

d extracts all the row elements from the 2nd and 3rd columns in c . The result is a 3 x 2 matrix

$$d = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ -1 & 0 \end{bmatrix}$$

Question: what is $c(2:3,1:2)$?



- `a = [1 2 3 4 5];` % 1x5 vector
- `b = [1;2;3;4;5];` % 5x1 vector
- `c = a';` %transpose of a
- `d = [1 2 3; 4 5 6; 7 8 9];` %3x3 matrix
- `e = [1:10];` % e=[1 2 3 4 5 6 7 8 9 10];
- `ee = [1:2:10];` % ee=[1 3 5 7 9];
- `f = zeros(5);` %5x5 zero matrix
- `g = ones(10);` % 10x10 matrix whose elements are all 1
- `h = eye(10);` %10x10 matrix whose diagonal elements are 1
- `i = rand(5,1);`
% a 5x1 vector whose elements are randomly generated
- `k = [];` % empty vector

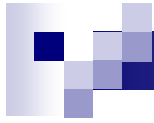


Commonly Used Matrix Operation

- $A+B$
- $A-B$
- $A*B$ $A.*B$
- A/B $A./B$
- A^b $A.^b$


$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$
$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$
$$A*B = \begin{bmatrix} 6 & 6 & 6 \\ 15 & 15 & 15 \\ 24 & 24 & 24 \end{bmatrix}$$
$$A.*B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$
$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$
$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$
$$A*B = \begin{bmatrix} 6 \\ 15 \\ 24 \end{bmatrix}$$
$$B*A = ?$$

X



Commands and Functions

- Use help command: `help function_name`
- help browser
- Functions - By Category
- Functions – Alphabetical List



Some commonly used functions

- | | | |
|-----------|---------|---------|
| ■ length | ■ eye | ■ sin |
| ■ size | ■ ones | ■ cos |
| ■ max | ■ zeros | ■ exp |
| ■ min | ■ rand | ■ log |
| ■ reshape | ■ mod | ■ log2 |
| ■ sort | ■ round | ■ log10 |
| ■ diag | ■ sign | ■ sqrt |
| | ■ floor | ■ abs |

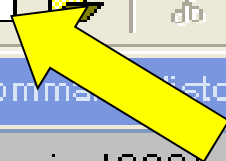


MATLAB Scripts/Codes

- A script file is an external file that contains a sequence of MATLAB statements
- By typing the filename, all statements in that file are executed sequentially
- The file should be in the current directory or MATLAB search path: `cd`, `pwd`
- Script files have extension `.m` called M-file
- `%` leads comment text



File Edit View Web Window Help



Current Directory: C:\MATLAB6p5\work

Command Window

```
magic(299)
```

```
A
```

```
C = cat(3,[1 2;3 4],[5 6;7 8])
```

```
magic(299)
```

```
B = C'
```

```
B = C+100;
```

```
help
```

```
helpbrowser
```

```
helpdesk
```

```
help helpdesk
```

```
search
```

```
help search
```

Press here to edit a
new file

Command Window

```
>> helpbrowser
```

```
>> helpdesk
```

```
>> help helpdesk
```

```
HELPDESK Comprehensive h  
HELPDESK displays the  
in the MATLAB Help br
```

```
>> search
```

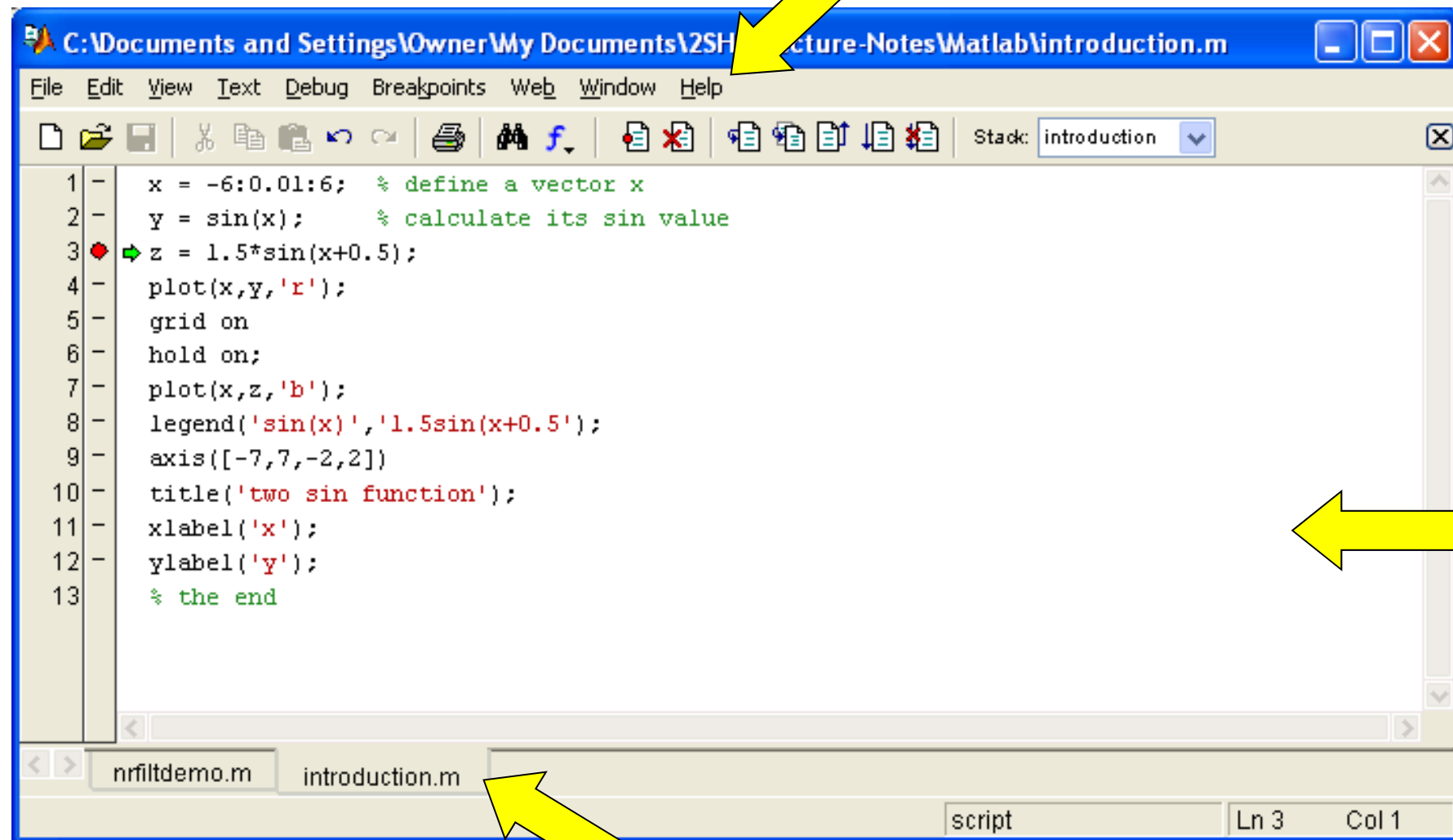
```
??? Undefined function or
```

```
>> help search
```

```
search.m not found.
```

```
>> help find
```

Editor





Programming with MATLAB

- Control flow
- Input/Output
- MATLAB function



if

```
if A == 3
    M = zeros(3, 3);
elseif A == 5
    M = zeros(5, 5);
else
    M = zeros(7, 7)
end
```




switch and case

```
swi tch A  
    case 10  
        ' 10'  
    case 20  
        ' 20'  
    case 30  
        ' 30'  
    otherwi se  
        ' 50'  
end
```



for

```
for n = 3:32  
    r(n) = n*(n-1);  
end
```

```
for i = 1:m  
    for j = 1:n  
        H(i,j) = 1/(i+j);  
    end  
end
```



while

```
b=10;  
X=0;  
while b >0  
    x = x+b;  
    b = b-1;  
end
```



break

```
x = 1;  
b = 1000;  
while b > 0  
    x = x*b;  
    if x > 9999999999999999  
        break  
    end  
end  
x
```



Input and Output

<code>fclose</code>	Close one or more open files
<code>feof</code>	Test for end-of-file
<code>ferror</code>	Query MATLAB about errors in file input/output
<code>fgetl</code>	Return next line as string without line terminator(s)
<code>fgets</code>	Return next line as string with line terminator(s)
<code>fopen</code>	Open file or obtain information about open files
<code>fprintf</code>	Write formatted data to file
<code>fread</code>	Read binary data from file
<code>frewind</code>	Rewind open file
<code>fscanf</code>	Read formatted data from file
<code>fseek</code>	Set file position indicator
<code>ftell</code>	Get file position indicator
<code>fwrite</code>	Write binary data to file

For more details, type “`help filename`” in the command window.



Functions

- Functions are M-files that can **accept** input arguments and **return** output arguments
- It's a good style to **make the name of the M-file and of the function be the same**
- Functions operate on variables within their own workspace (local variables)
- When call a MATLAB function, the corresponding M-file should be in the **current directory** or MATLAB **search directory**



■ Example

```
function C = add(A,B)
```

```
% Add two Matrices.
```

```
[na,ma]=size(A);
```

```
[nb,mb]=size(B);
```

```
If (na==nb)&(ma==mb)
```

```
    C=A+B;
```

```
else
```

```
    fprintf('The two matrices cannot be added.\n');
```

```
end
```



Graphics

- plot
- xlabel
- ylabel
- title
- legend
- subplot
- axis

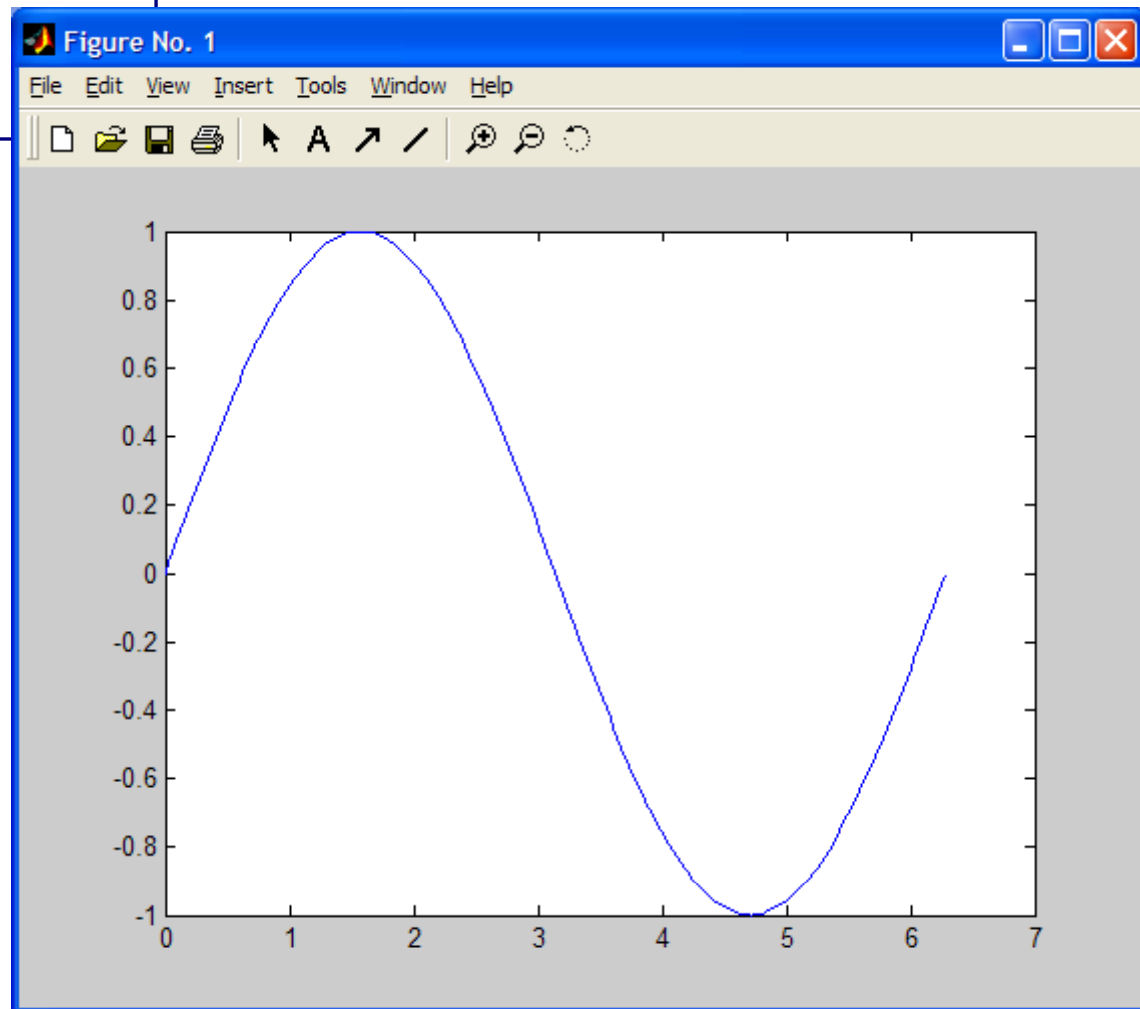


plot

- Draw a piecewise linear graph
- `plot(y):` $y(i)$ vs. i
- `plot(x,y):` $y(i)$ vs. $x(i)$

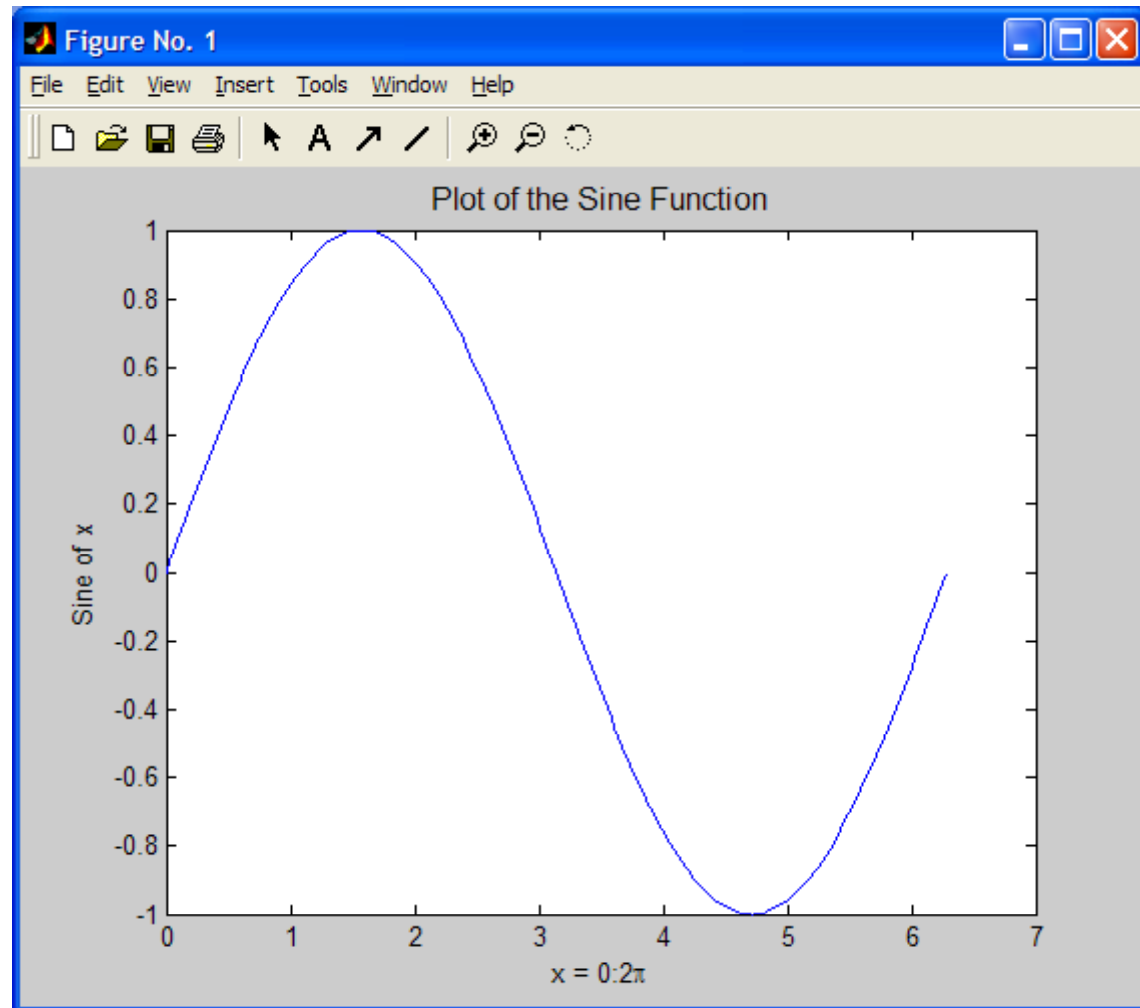
Example: plot a sine wave.

```
x = 0: pi /100: 2*pi ;  
y = sin(x);  
plot(x, y)
```



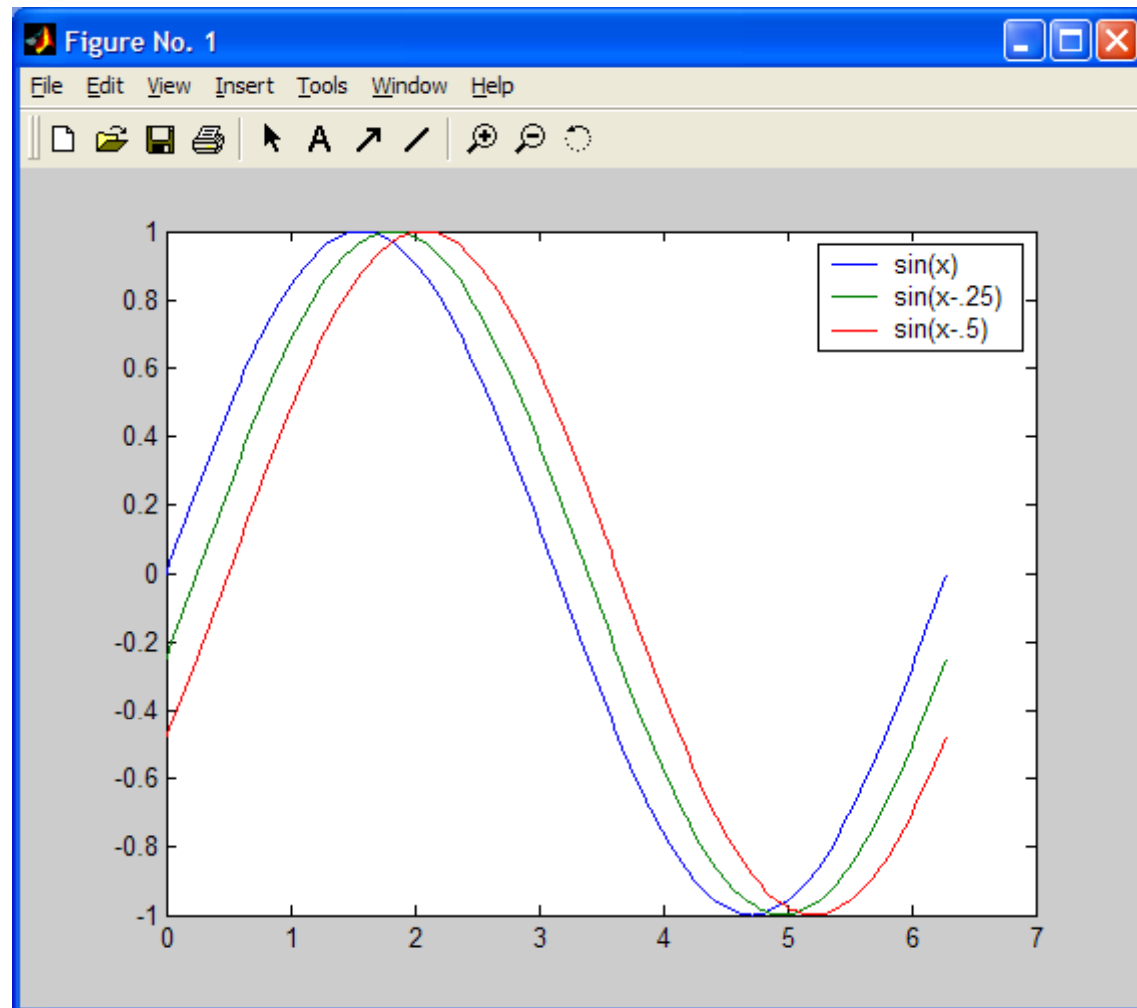
```
xlabel('x = 0:2\pi')  
ylabel('Sine of x')  
title('Plot of the Sine Function', 'FontSize', 12)
```

Add labels
and title.



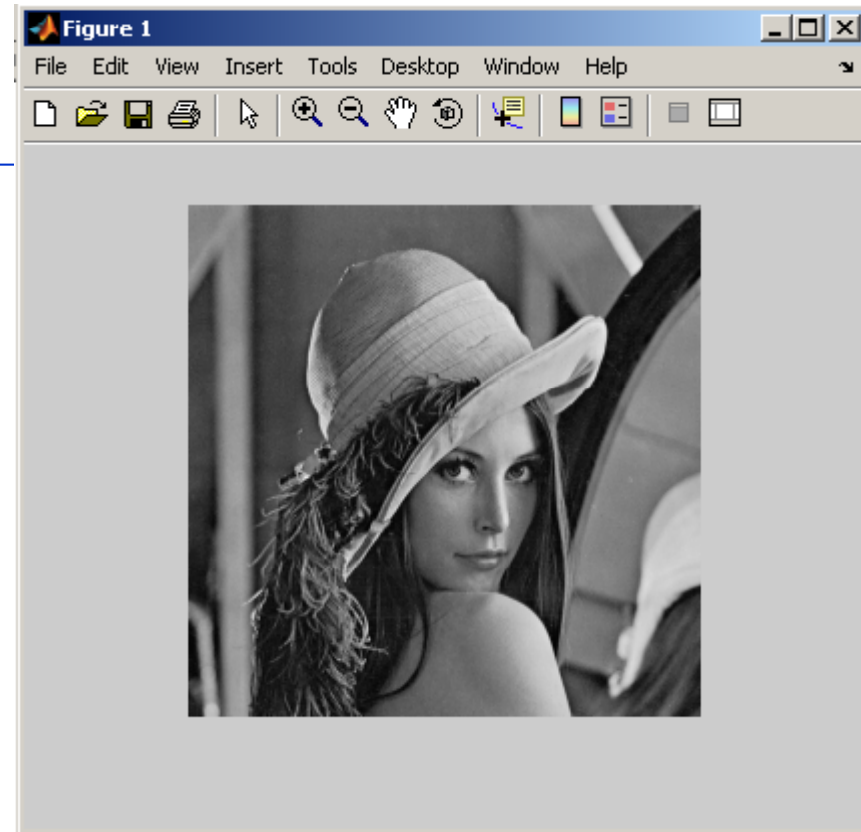
```
y2 = sin(x-.25);  
y3 = sin(x-.5);  
plot(x, y, x, y2, x, y3)  
legend('sin(x)', 'sin(x-.25)', 'sin(x-.5)')
```

Plot several
curves in one
figure.



To read and show an gray image

```
clear;  
%clear the workspace  
  
I = imread('lena', 'tif');  
%read the lena.tif file  
  
I = double(I);  
%convert to double precision  
  
figure(1);  
%open a figure window  
imshow(I, [0 255]);  
%show the image with gray level [0 255]
```



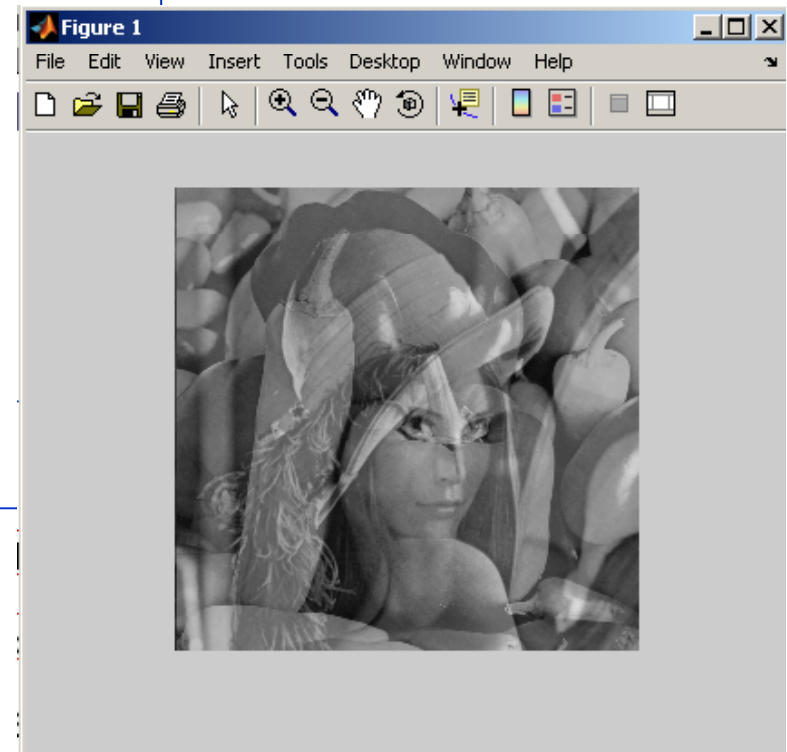
To write an gray image

```
clear;

I1=double(imread('lena','tif'));
%read the lena.tif file
I2=double(imread('peppers','tif'));
%read the peppers.tif file

I=(I1+I2)/2;
%average the two images
figure(1);imshow(I, [0 255]);

imwrite(I/255,'test.tif','tif');
%write the image to "test.tif"
```

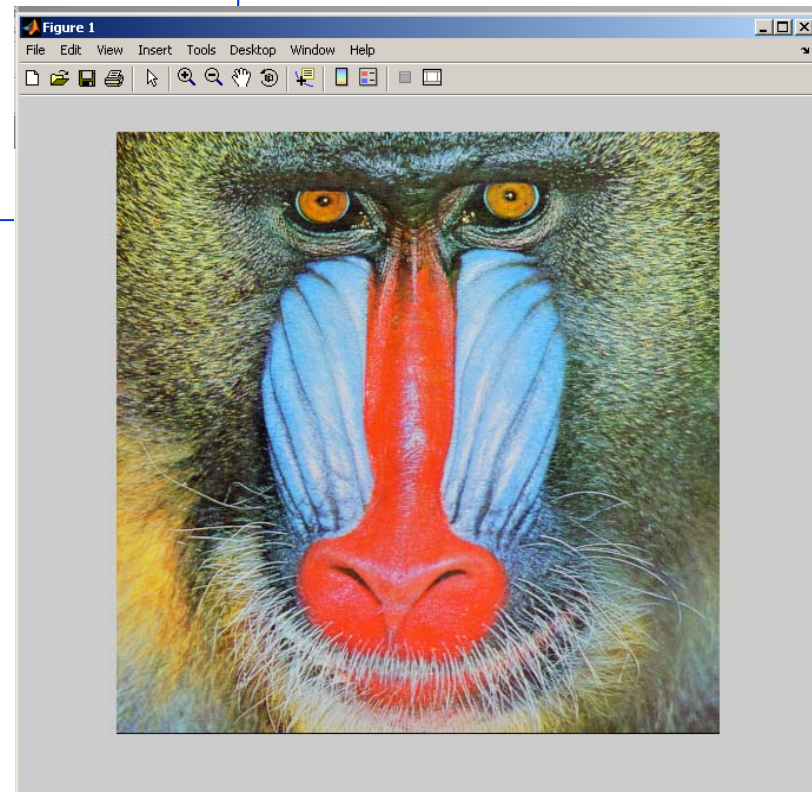


To read a color image

```
clear;
```

```
I = double(imread('baboon_c', 'tif'));  
%read the baboon_c.tif file
```

```
figure(1);  
imshow(I/255); %show the image
```



To write a color image

```
clear; %clear the workspace
```

```
I1=double(imread('baboon_c','tif'));  
%read the baboon_c.tif file
```

```
I2=double(imread('peppers_c','tif'));  
%read the peppers_c.tif file
```

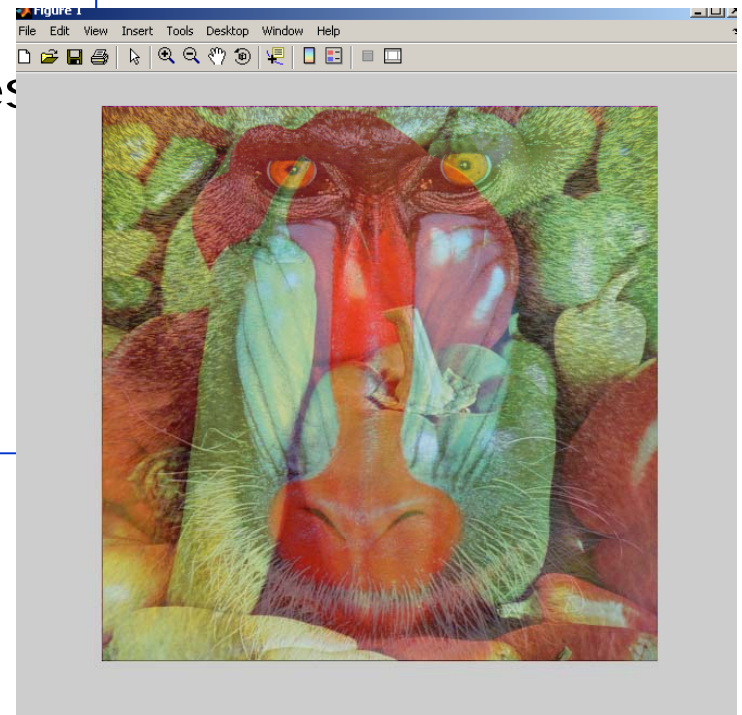
```
I=(I1+I2)/2; %average the two images
```

```
figure(1);imshow(I/255);
```

```
%show the image
```

```
imwrite(I/255,'test_c.tif','tif');
```

```
%write the image to "test_c.tif"
```



Example: transition

- How would you create your own video wipe transition from the top-left corner of the viewport down to the bottom-right corner -- a **diagonal transition**?



Matlab Codes: transition.m

```
■ clear all;%clear the workspace
■ close all; %close all windows
■ im1=double(imread('lenna256_c','tif'));
  %read the lenna image
■ im2=double(imread('baboon256_c','tif'));
  %read the baboon image
■ figure(1),clf;
■ imshow(im1/255);
■ figure(2),clf;
■ imshow(im2/255);
■
■ [ymax, xmax, c] = size(im1);
■ %get the size of im1
■ %ymax is the row number, xmax is the
  %column number and c=3 for color
  %images
■ m = ymax/xmax;

■ tmax = 1;
■ dmax =sqrt(xmax*xmax+ymax*ymax);
■ out = zeros(ymax,xmax,3);
■
■ for t=0:0.1:tmax
■     dtrans = dmax*t/tmax;
■     for y=1:ymax
■         for x=1:xmax
■             d = dmax/2 * (y+m*x)/ymax;
■             if d < dtrans
■                 out(y,x,:) = im1(y,x,:);
■             else
■                 out(y,x,:) = im2(y,x,:);
■             end
■         end
■     end
■ end
■ figure(3),clf;
■ imshow(out/255);
■ end
```



Read audio files

- `wav=wavread('Windows XP_wav.wav');` % read a audio file
- `wavplay(wav);` % play the audio file



Read video files

- `avi = aviread('ellipse.avi');% read a video file`
- `figure(1),clf;`
- `movie(avi);%play`



End