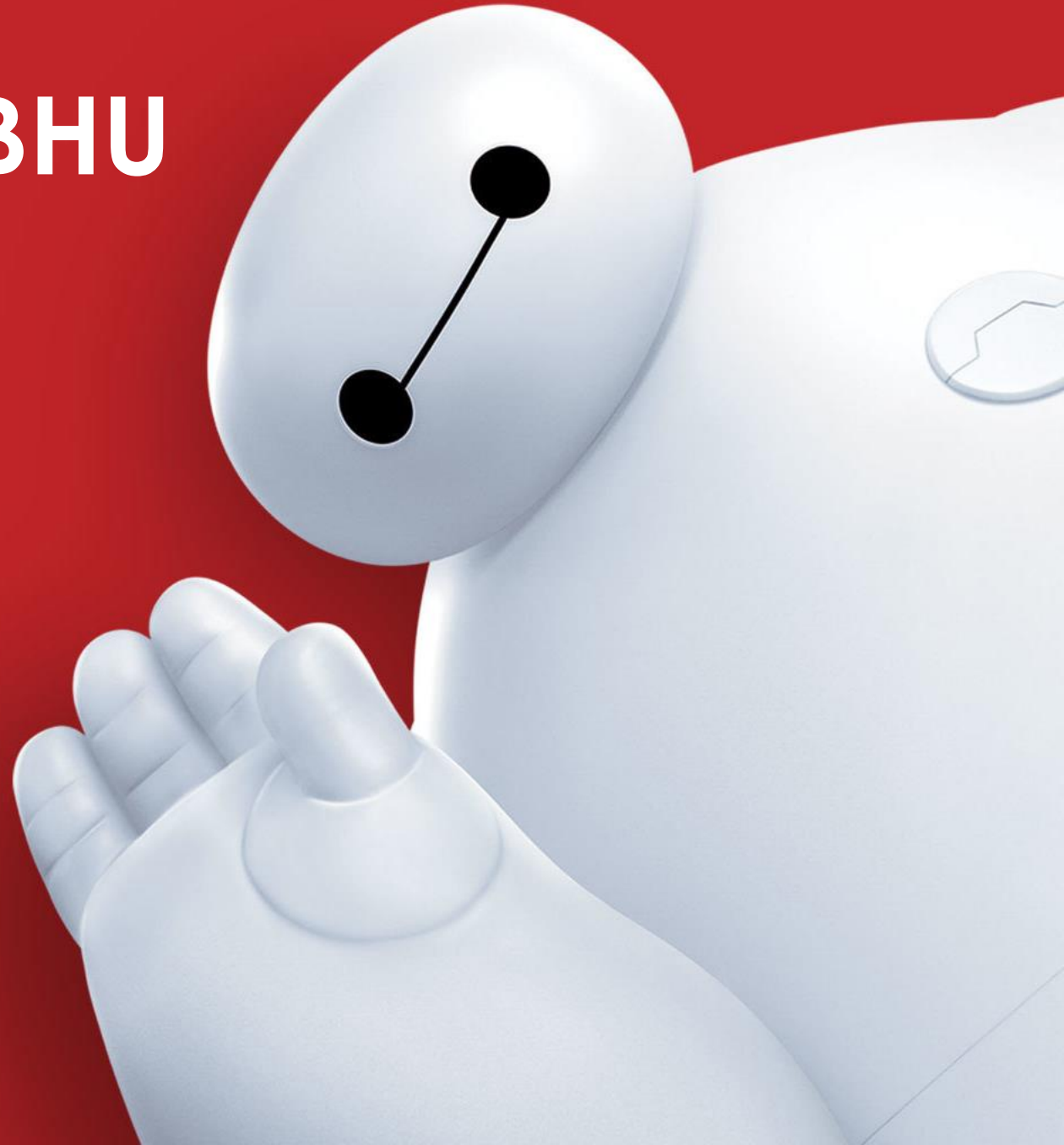


# Robotics Club, IIT BHU

Presents

## Introduction to Computer Vision



```
graph TD; A[Topics to be covered] --> B[Basics of MATLAB]; A --> C[Image Processing]; A --> D[Video Processing];
```

Topics to be covered

Basics of  
MATLAB

Image  
Processing

Video  
Processing

# Basics of MATLAB:



# Features of MATLAB:

- ▶ MATLAB (matrix laboratory) is a fourth-generation high-level programming language .
- ▶ It also provides an interactive environment for iterative exploration, design and problem solving.
- ▶ Image analysis, including segmentation, morphology, statistics, and measurement.

# Uses of MATLAB:

- ▶ Image and Video Processing.
- ▶ Signal Processing and Communications.
- ▶ Test and Measurement.
- ▶ Control Systems.
- ▶ Computational Finance.

# Know Your MATLAB Window:

The image displays the MATLAB R2013a software interface. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below the menu bar is a toolbar with various icons for file operations and a search bar labeled 'Search Documentation'. The main workspace is divided into four panes:

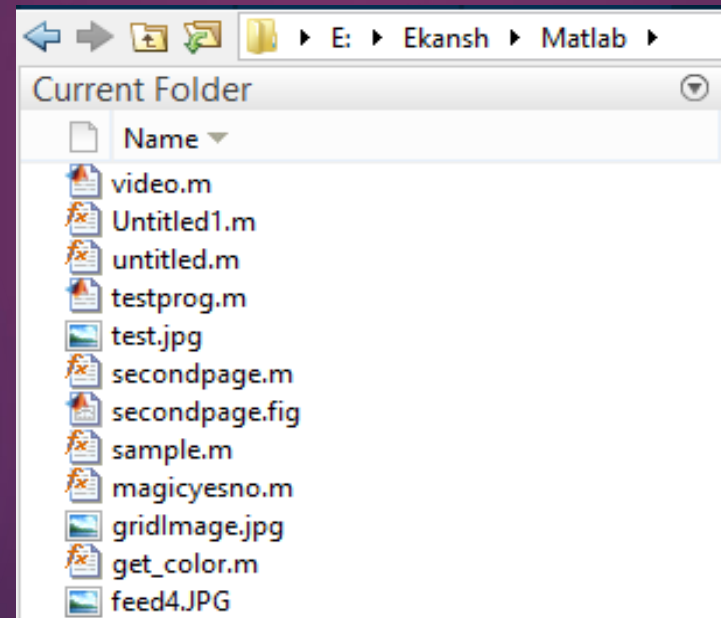
- Current Folder:** A file explorer showing the contents of the 'E:\Ekansh\Matlab' directory. It lists files such as 'video.m', 'Untitled1.m', 'untitled.m', 'testprog.m', 'test.jpg', 'secondpage.m', 'secondpage.fig', 'sample.m', 'magicyesno.m', 'gridImage.jpg', 'get\_color.m', 'feed4.JPG', 'extract\_color.m', 'arena\_rgb.jpg', 'arena.png', 'arena.jpg', 'add.m', 'abcd.m', 'abc.png', 'New folder', and 'LAB'.
- Command Window:** A text area for entering MATLAB commands. It shows the following sequence of commands and their outputs:

```
>> a=4
a =
    4
>> b=8
b =
    8
>> c=a+b
c =
   12
fx >>
```
- Workspace:** A table displaying the current workspace variables. It has columns for 'Name', 'Value', 'Min', and 'Max'. The variables shown are 'a' (value 4), 'b' (value 8), and 'c' (value 12).
- Command History:** A list of previously executed commands, including:

```
rand(5)
rand()
2*randint(1,10)-1
help conv
t = [-1:0.05:1];
x = 4*sin(2*pi*t);
help imshow
help imtool
help imaqtool
clear all
clc
a=4
b=8
c=a+b
```

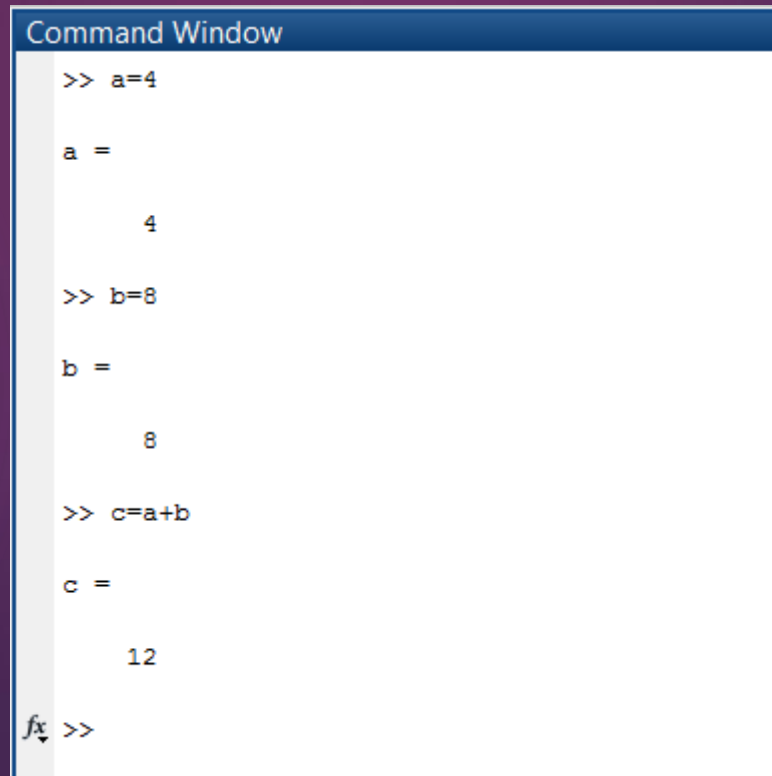
# Know Your MATLAB Window:

- ▶ Current Folder: This panel allows you to access the project folders and files.



# Know Your MATLAB Window:

- ▶ Command Window: This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).

A screenshot of the MATLAB Command Window. The window has a blue title bar that says "Command Window". The main area is white and contains the following text: ">> a=4", "a =", "4", ">> b=8", "b =", "8", ">> c=a+b", "c =", "12", and at the bottom, a cursor icon followed by ">>".

```
Command Window

>> a=4

a =

    4

>> b=8

b =

    8

>> c=a+b

c =

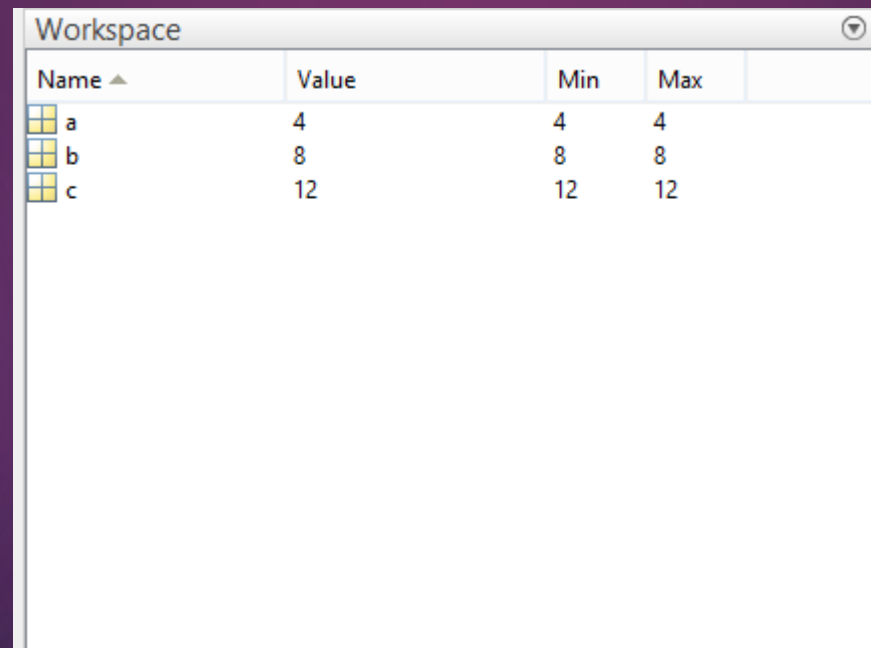
   12

fx >>
```



# Know Your MATLAB Window:

- ▶ Workspace: The workspace shows all the variables created and/or imported from files.

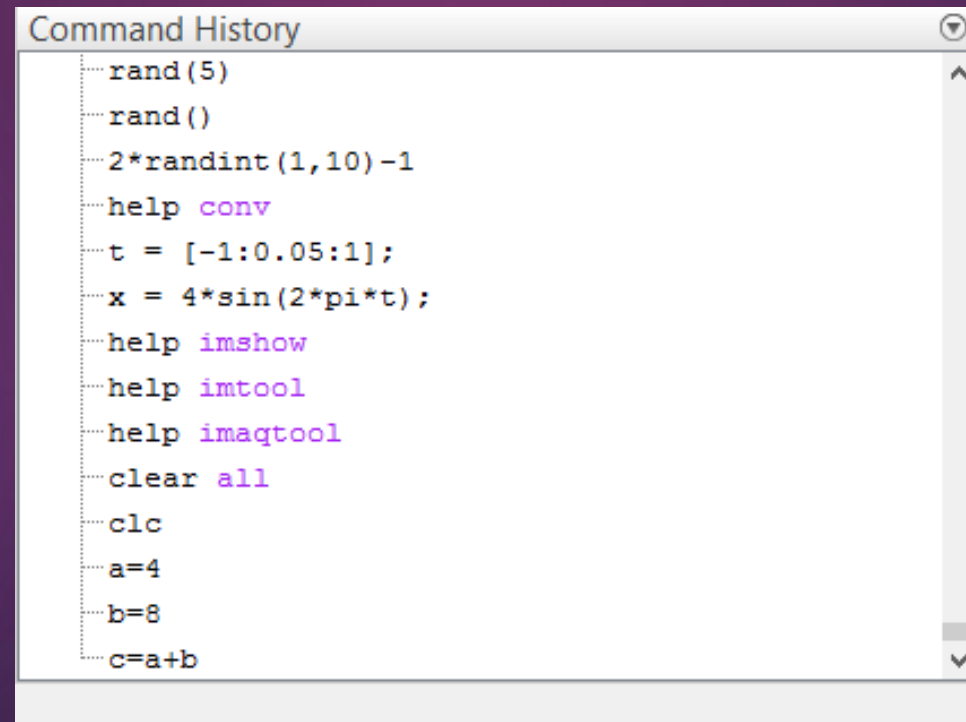


The screenshot shows the MATLAB Workspace window. It has a title bar labeled 'Workspace' with a dropdown arrow on the right. Below the title bar is a table with four columns: 'Name', 'Value', 'Min', and 'Max'. There are three rows of data, each preceded by a small yellow square icon with a blue 'x' in the top-left corner. The first row is for variable 'a' with a value of 4, min of 4, and max of 4. The second row is for variable 'b' with a value of 8, min of 8, and max of 8. The third row is for variable 'c' with a value of 12, min of 12, and max of 12.

Name ▲	Value	Min	Max
a	4	4	4
b	8	8	8
c	12	12	12

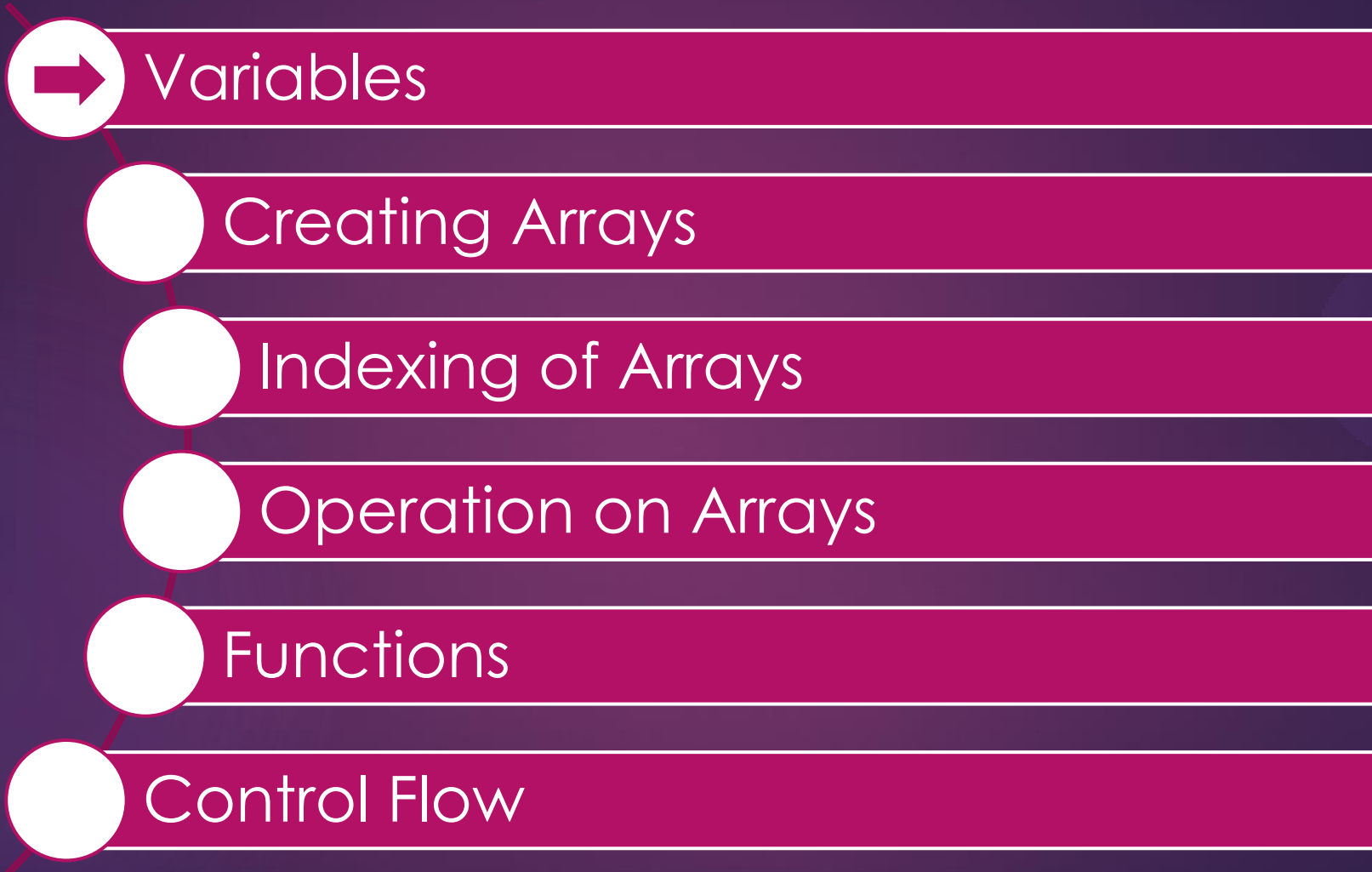
# Know Your MATLAB Window:

- ▶ Command History: This panel shows or rerun commands that are entered at the command line.

A screenshot of the MATLAB Command History window. The window has a title bar that says "Command History" and a close button. The main area is a list of commands entered at the command line, each preceded by a small icon. The commands are: rand(5), rand(), 2\*randint(1,10)-1, help conv, t = [-1:0.05:1];, x = 4\*sin(2\*pi\*t);, help imshow, help imtool, help imaqtool, clear all, clc, a=4, b=8, and c=a+b. The window has a scrollbar on the right side.

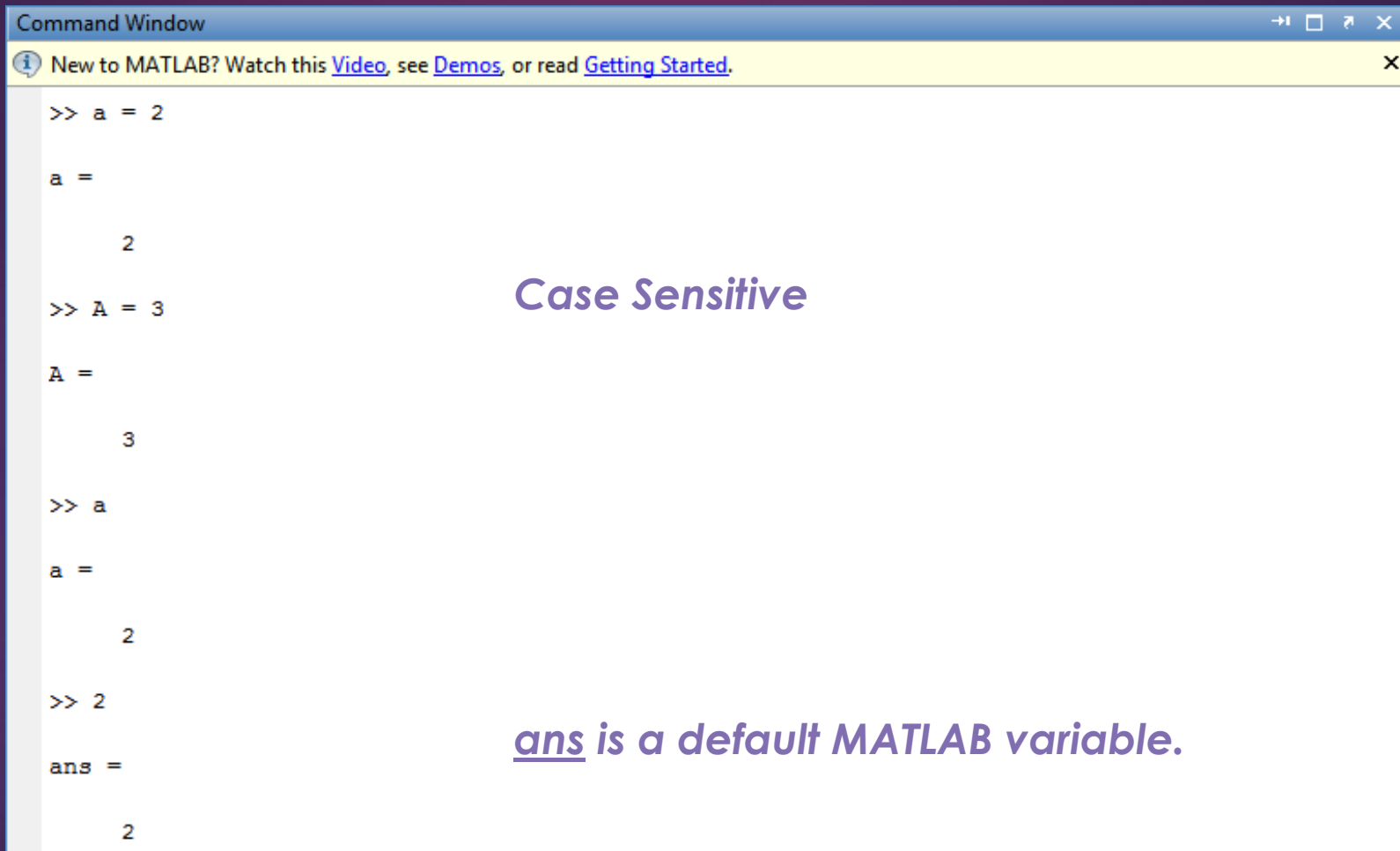
```
Command History
rand(5)
rand()
2*randint(1,10)-1
help conv
t = [-1:0.05:1];
x = 4*sin(2*pi*t);
help imshow
help imtool
help imaqtool
clear all
clc
a=4
b=8
c=a+b
```

# Basics of MATLAB:



# Basics of MATLAB:

## ► Variable:



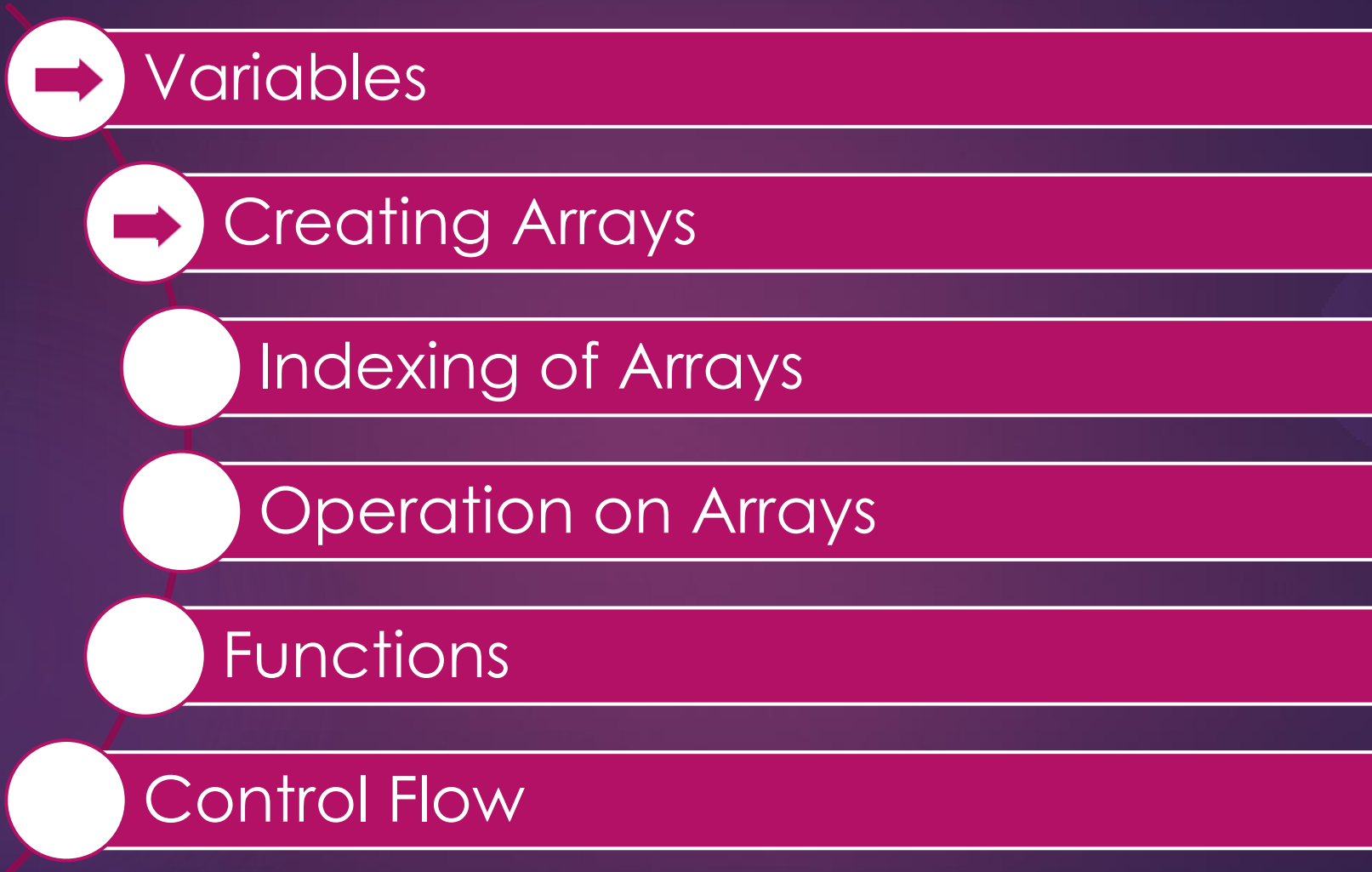
The screenshot shows the MATLAB Command Window with the following text:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a = 2
a =
    2
>> A = 3
A =
    3
>> a
a =
    2
>> 2
ans =
    2
```

*Case Sensitive*

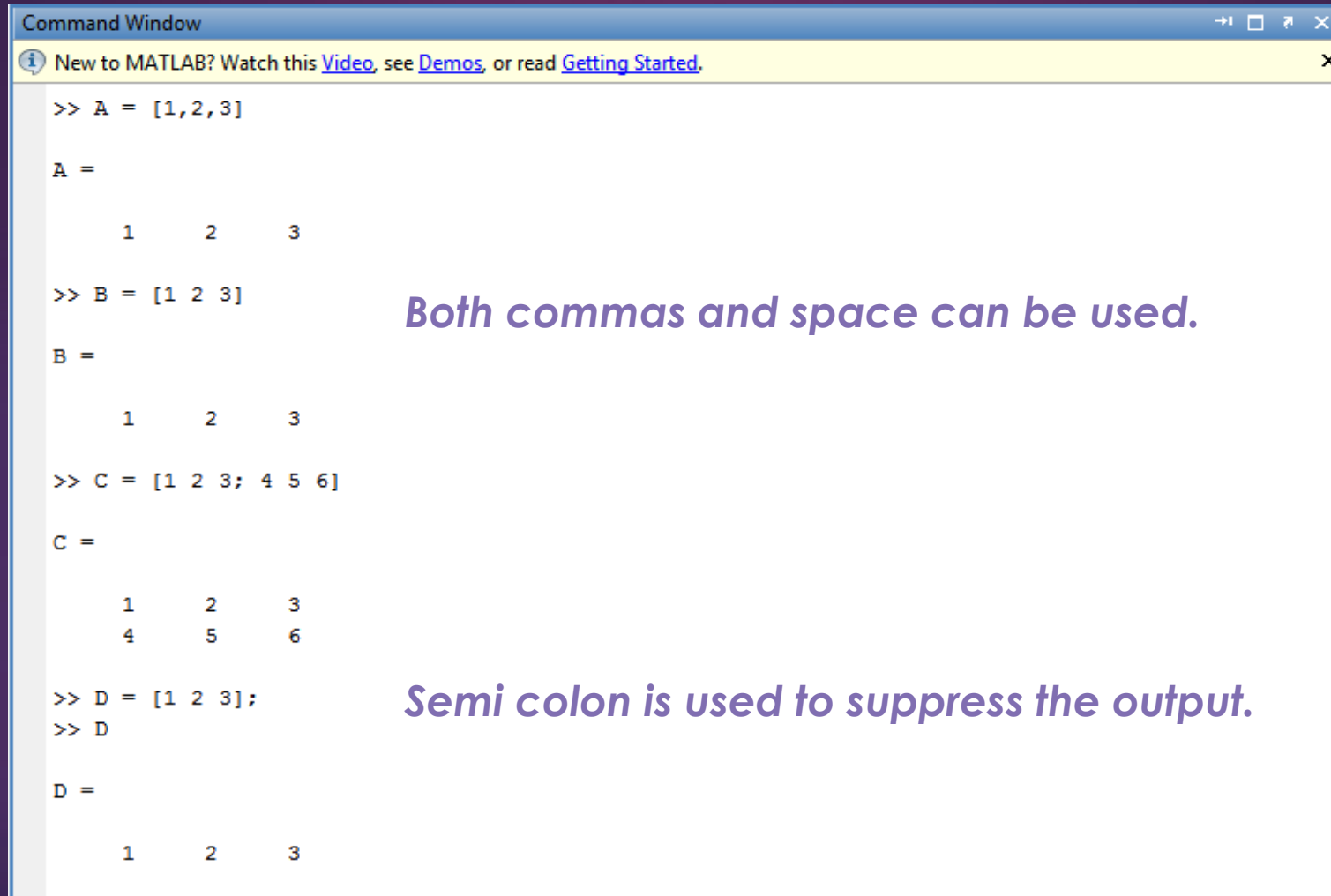
*ans is a default MATLAB variable.*

# Basics of MATLAB:



# Basics of MATLAB:

## ► Creating Arrays:



The screenshot shows the MATLAB Command Window with the following content:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> A = [1,2,3]

A =

     1     2     3

>> B = [1 2 3]

B =

     1     2     3

>> C = [1 2 3; 4 5 6]

C =

     1     2     3
     4     5     6

>> D = [1 2 3];
>> D

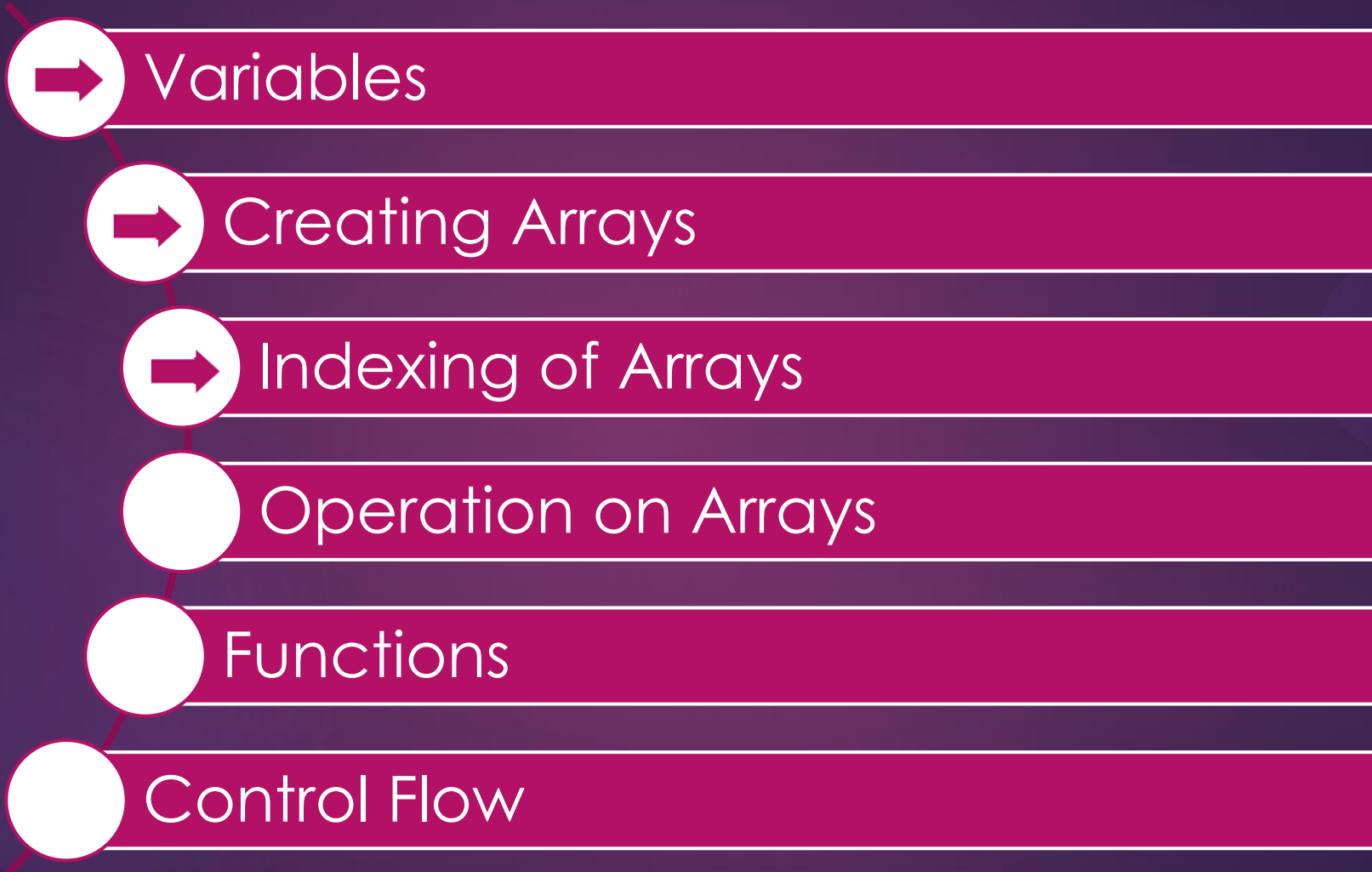
D =

     1     2     3
```

*Both commas and space can be used.*

*Semi colon is used to suppress the output.*

# Basics of MATLAB:



# Basics of MATLAB:

## ► Indexing of Arrays:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> D = [1 2 3; 4 5 6; 7 8 9]

D =

     1     2     3
     4     5     6
     7     8     9

>> D(2,3)

ans =

     6

>> D(1:2, 2)

ans =

     2
     5

>> D(1:2, 2:3)

ans =

     2     3
     5     6
```

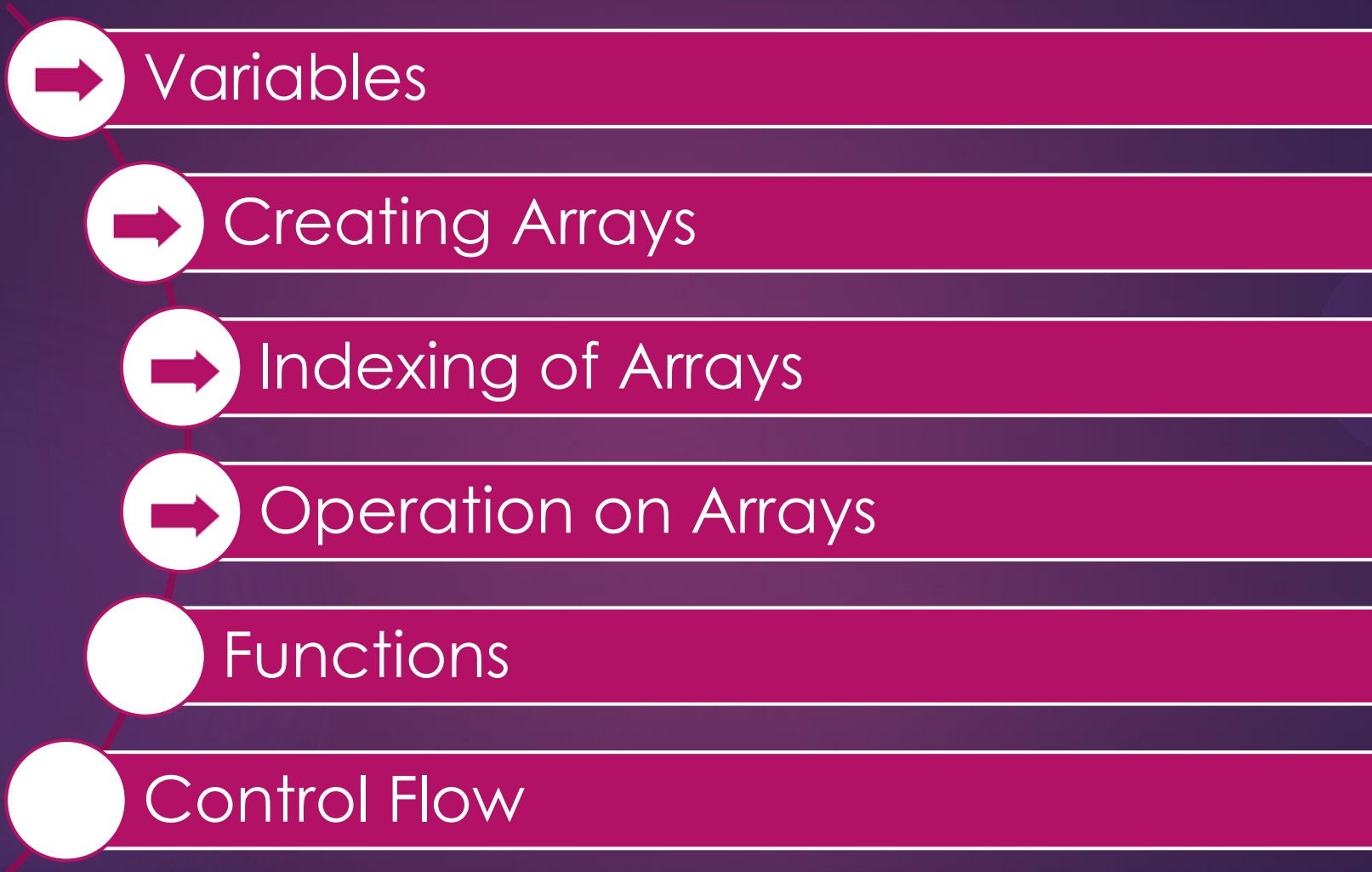
*Observe that the first element is (1,1) and not (0,0).*

*Colon operator is used for traversal.*

*Check the size using: size()*



# Basics of MATLAB:



# Basics of MATLAB:

## ► Operations on Arrays:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> D

D =

     1     2     3
     4     5     6
     7     8     9

>> D + D

ans =

     2     4     6
     8    10    12
    14    16    18

>> D - D

ans =

     0     0     0
     0     0     0
     0     0     0

>> D * D

ans =

    30    36    42
    66    81    96
   102   126   150
```

*Matrix + Matrix*

*Matrix - Matrix*

*Matrix \* Matrix*

# Basics of MATLAB:

## ► Operations on Arrays:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> D + 2
ans =
     3     4     5
     6     7     8
     9    10    11

>> D - 2
ans =
    -1     0     1
     2     3     4
     5     6     7

>> D * 2
ans =
     2     4     6
     8    10    12
    14    16    18

>> D / 2
ans =
    0.5000    1.0000    1.5000
    2.0000    2.5000    3.0000
    3.5000    4.0000    4.5000
```

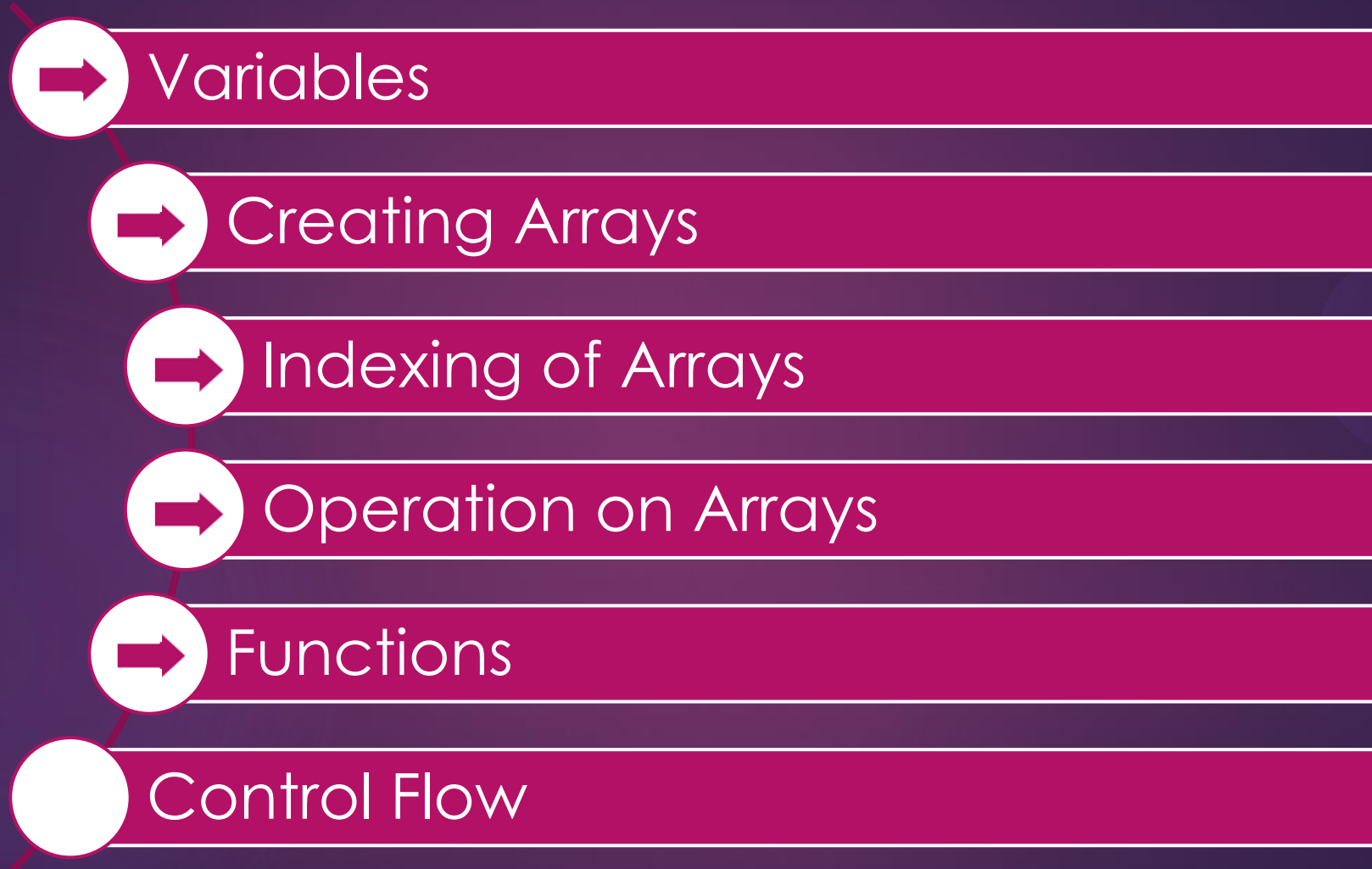
*Matrix + Scalar*

*Matrix - Scalar*

*Matrix \* Scalar*

*Matrix / Scalar*

# Basics of MATLAB:

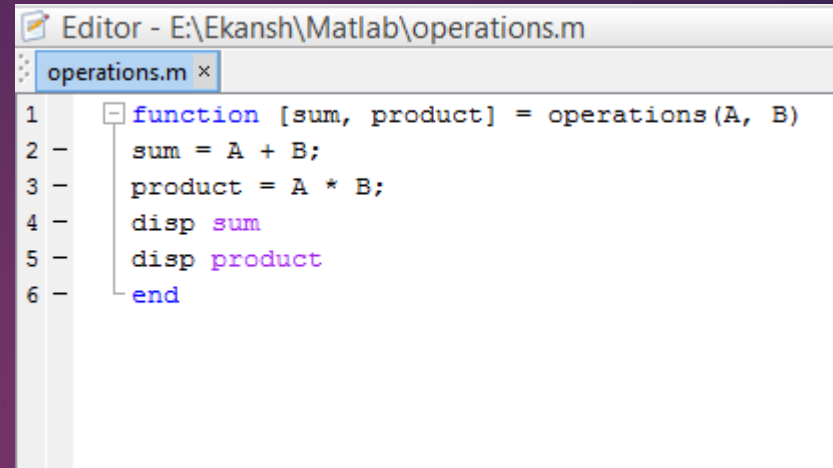


# Basics of MATLAB:

## ► Functions:

### ► To open the editor:

- File->New->Script
- Ctrl + N
- New Blank File Icon



The screenshot shows the MATLAB Editor window with the file 'operations.m' open. The code is as follows:

```
1 function [sum, product] = operations(A, B)
2     sum = A + B;
3     product = A * B;
4     disp sum
5     disp product
6 end
```

### ► Function header syntax:

function [output variables] = function\_name (input\_variables)

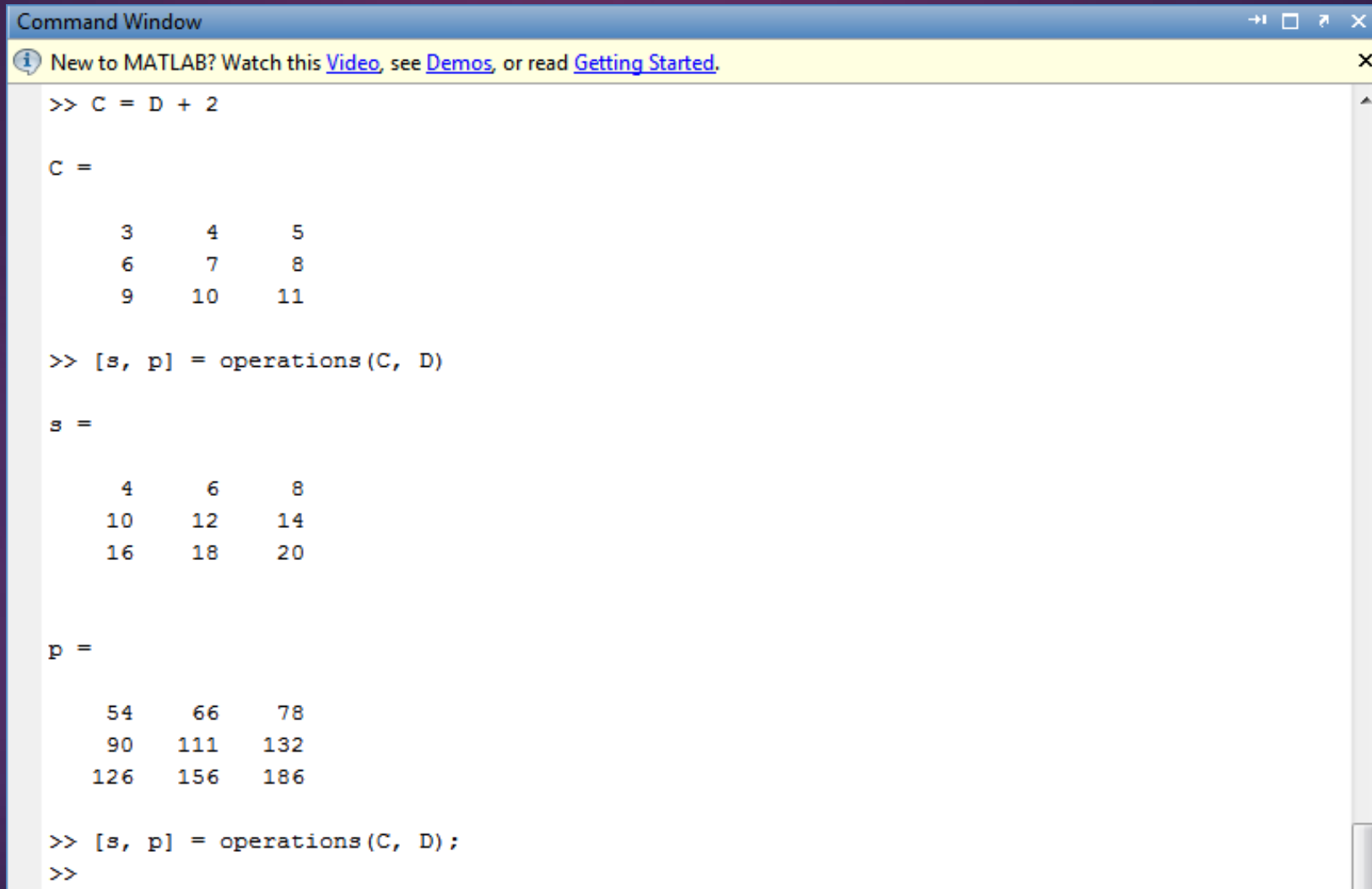
eg.

function[sum, product] = operations(A, B)

- Remember that the function name and the 'm-file' name *MUST* be the same.

# Basics of MATLAB:

## ► Functions:



The screenshot shows the MATLAB Command Window interface. At the top, there is a title bar 'Command Window' and a yellow information banner that reads: 'New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).' Below the banner, the command history is displayed. The first command is `>> C = D + 2`, which results in the variable `C` being assigned a 3x3 matrix of values: 3, 4, 5; 6, 7, 8; 9, 10, 11. The second command is `>> [s, p] = operations(C, D)`, which results in two variables, `s` and `p`, being assigned 3x3 matrices. The matrix `s` contains the values: 4, 6, 8; 10, 12, 14; 16, 18, 20. The matrix `p` contains the values: 54, 66, 78; 90, 111, 132; 126, 156, 186. The third command is `>> [s, p] = operations(C, D);`, which is a second call to the same function. The window ends with `>>`.

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> C = D + 2

C =

     3     4     5
     6     7     8
     9    10    11

>> [s, p] = operations(C, D)

s =

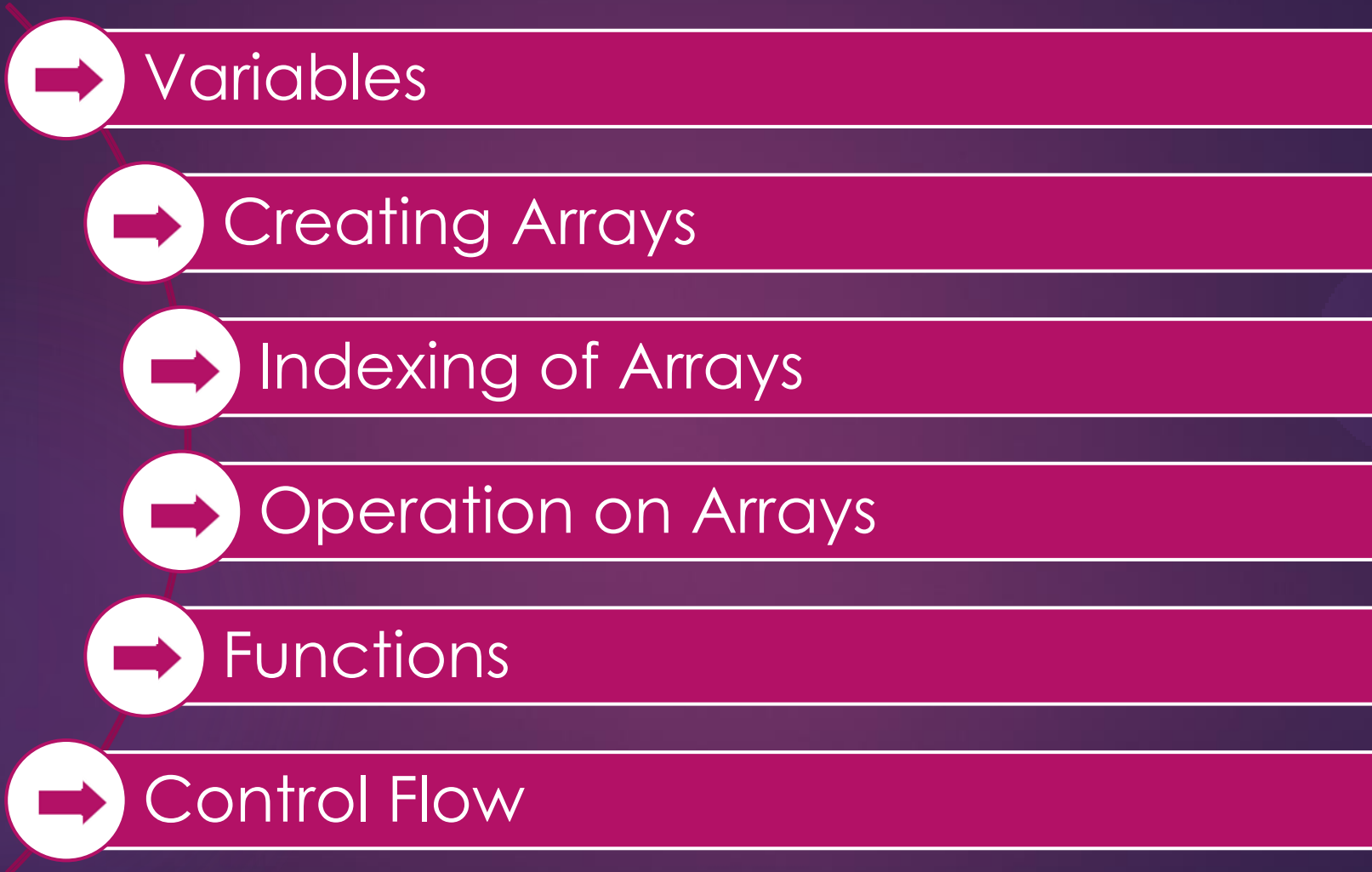
     4     6     8
    10    12    14
    16    18    20

p =

    54    66    78
    90   111   132
   126   156   186

>> [s, p] = operations(C, D);
>>
```

# Basics of MATLAB:



# Basics of MATLAB:

## ► Control Flow Statements:

```
Editor - E:\Ekansh\Matlab\control_flow.m
control_flow.m x
1 function [] = control_flow()
2     i = 0 ; j = 0;
3
4     disp('while loop');
5     while i < 5
6         disp (i)
7         i = i + 1;
8     end
9
10    disp('for loop');
11    for j = 0:3:9
12        disp (j)
13    end
14
15    disp('if elseif statement');
16    if j == 9
17        disp ('Completely Closed Interval');
18    elseif j == 8
19        disp ('Semi Closed Interval');
20    else
21        disp ('j is neither 8 nor 9');
22    end
23    end
24
```

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started

while loop
    0
    1
    2
    3
    4

for loop
    0
    3
    6
    9

if else if statement
Completely Closed Interval
fx >> |
```



# Basics of MATLAB:

## ► A Few Commands:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> who

Your variables are:

A      B      C      D      a      ans     b      c      im      im1     p      s

>> whos

Name      Size      Bytes  Class  Attributes

A          1x3         24  double
B          1x3         24  double
C          3x3        72  double
D          3x3        72  double
a          1x1          8  double
ans        3x3        72  double
b         13x13       1352 double
c          1x5         40  double
im       287x250x3    215250 uint8
im1       75x100x3    22500  uint8
p          3x3         72  double
s          3x3         72  double

>> clear
>> who
>> whos
fx >> |
```

The command clc clears the screen of the command window

The command help helps. A LOT.

# Basics of MATLAB:

## ▶ A Few Functions:

- ▶ Trigonometric: `sin()`, `sind()`, `cos()`, `cosd()`, etc.
- ▶ Inverse trigonometric: `asin()`, `asind()`, `acos()`, `acosd()`, etc.
- ▶ `min()`, `max()`
- ▶ `size()`
- ▶ `sort()`
- ▶ `zeros()`
- ▶ `ones()`
- ▶ `eye()`



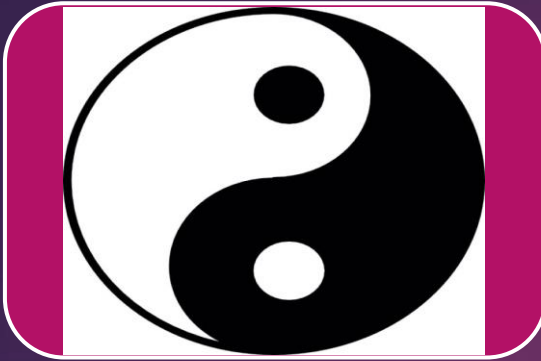
*...because every picture  
tells a story...*

# Image Processing:

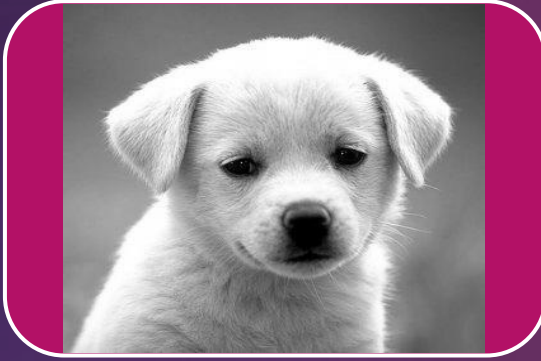
- ▶ Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it.
- ▶ Image processing basically includes the following three steps:



# Images:



- Binary Image



- Grayscale Image



- RGB Image

# Images in MATLAB:

- ▶ Each image is seen as a matrix of size equal to the number of pixel rows x number of pixel columns.
- ▶ Each pixel has a value of intensity.
- ▶ Each element of the matrix contains the value of this intensity at the corresponding to the pixel it represents.

# Images in MATLAB:

## Binary Image

- All the elements of the matrix are either zero or one.
- Zero represents black and 1 represents white.

## Grayscale Image

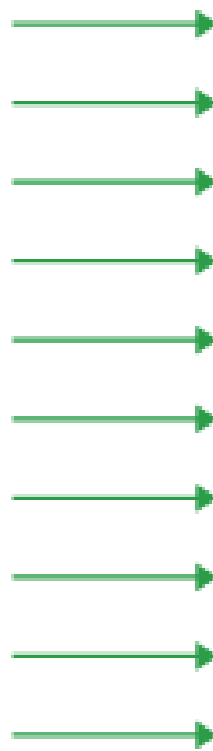
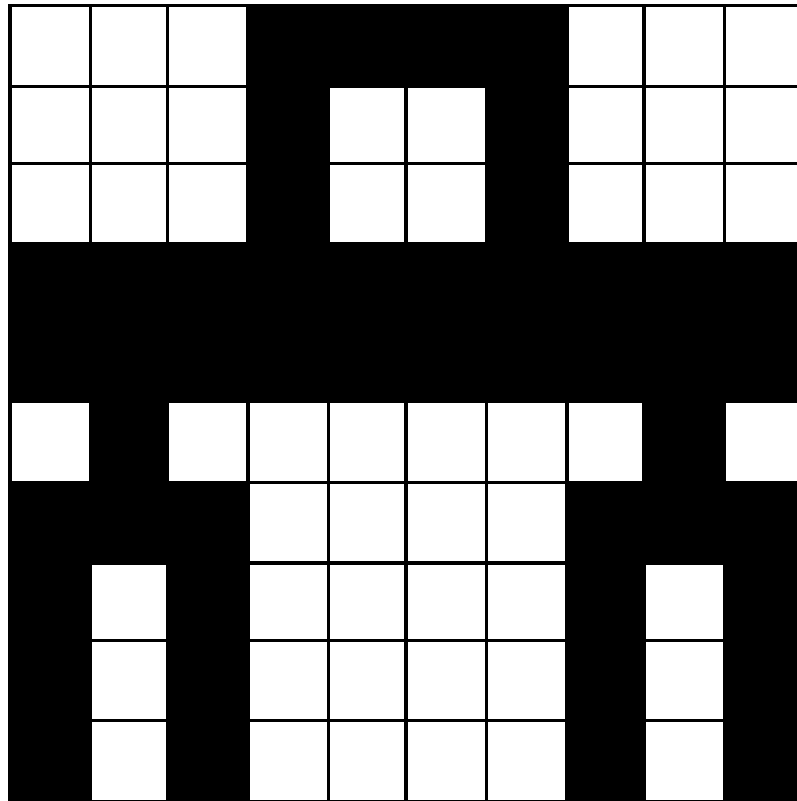
- All the elements of the matrix lie between 0 and 255.
- Zero represents Black, 255 represents White and the intermediate values represent shades of Gray.

## RGB Image

- Each colour has a specific RGB value!
- RGB Images are seen as 3D matrices with the 1<sup>st</sup> plane corresponding to R, 2<sup>nd</sup> to G and 3<sup>rd</sup> to B.

# Images in MATLAB:

## ► Binary Image:

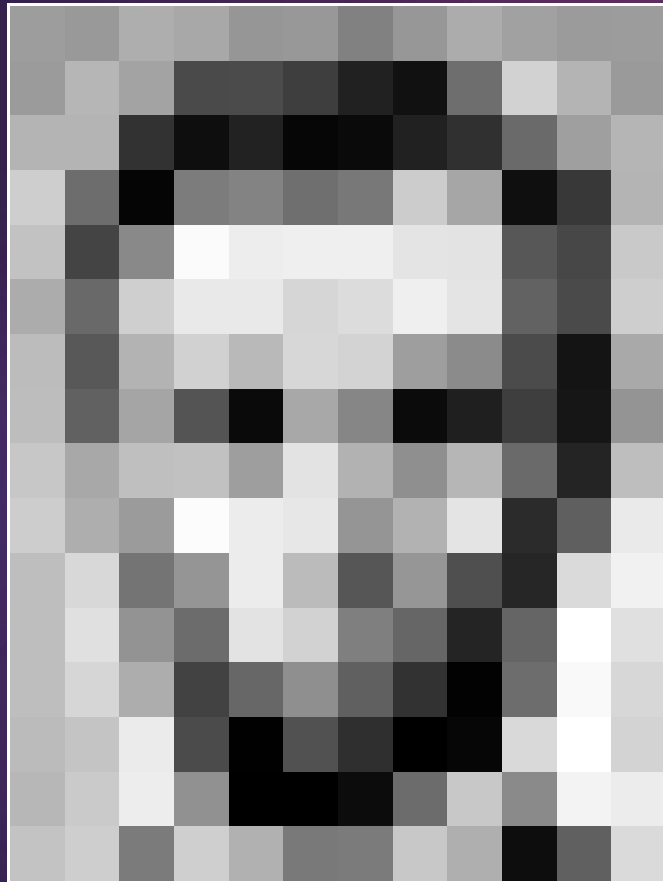


0	0	0	1	1	1	1	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	1	1	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1



# Images in MATLAB:

## ► Grayscale Image:



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	55	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	163	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	55	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	163	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

# Images in MATLAB:

## ► RGB Image:

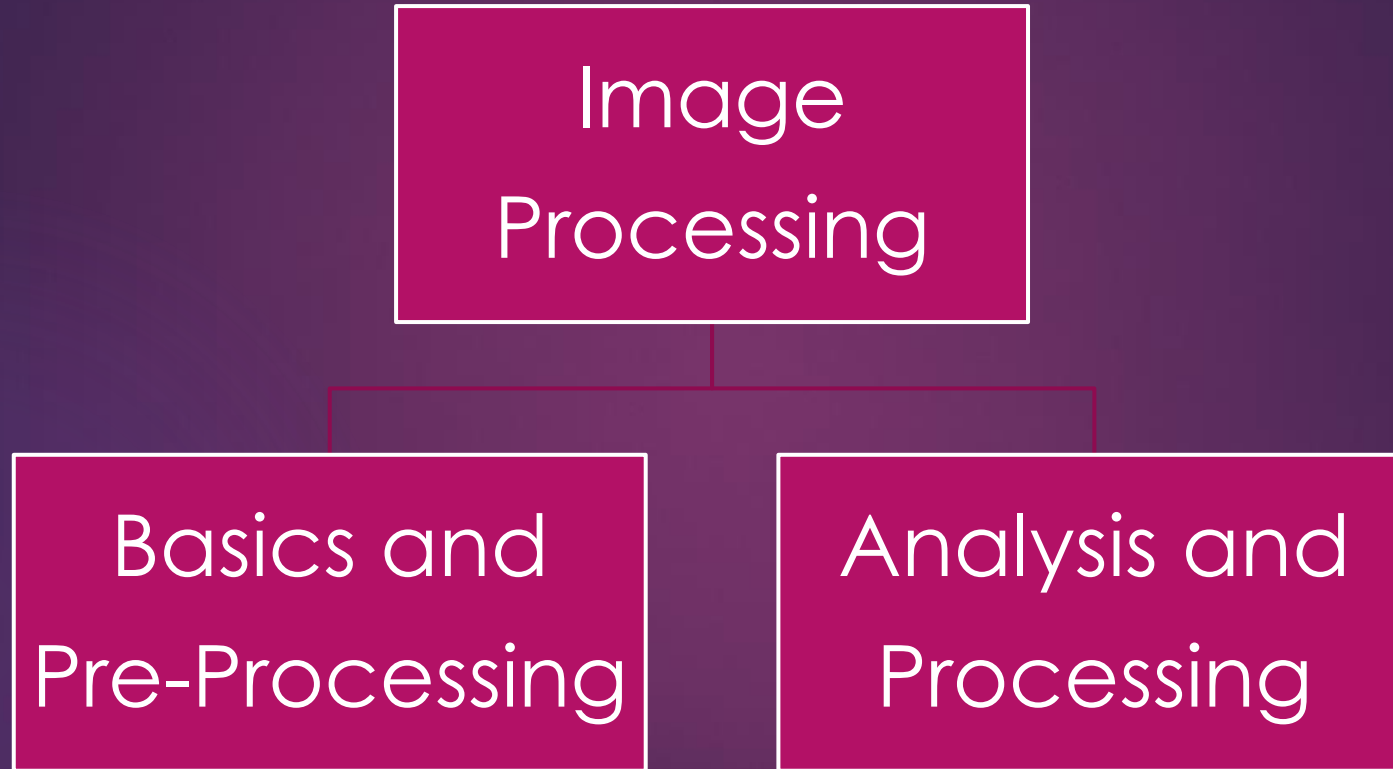
- Each pixel contains a vector representing red, green and blue components



RGB Components

10	10	16	28
9	65	70	56
15	32	99	70
32	21	60	90
	54	85	85
		32	65

# Image Processing:



# Basics and Pre-Processing

- ▶ `imread()`

Reads the image as a matrix.

- ▶ `imshow()`

Shows the image as an 'image'.

- ▶ `imtool()`

Same as `imshow` but with different tools.

# Basics and Pre-Processing

The input argument: File name along with extension.

The input argument: A Matrix.

Command Window

```
>> im = imread('bob.jpg');  
>> imshow(im)
```

```
Warning: Image is too big to  
> In imuitools\private\initSi  
In imshow at 283
```

```
fx >>
```

Figure 1



# Basics and Pre-Processing

## ▶ `imcrop()`

- ▶ `imcrop(im)` : Interactive crop tool
- ▶ `imcrop(im, [x1,y1,x2,y2])`

## ▶ `imresize()`

- ▶ `imresize(im, 0.5)`
- ▶ `imresize(im, [200, 200])`
- ▶ `imresize(im, [200, NaN] )` or `imresize(im, [NaN, 200] )`
- ▶ `imresize(im, 0.5, 'nearest')` or `imresize(im, 0.5, 'bilinear')` or `imresize(im, 0.5, 'bicubic')`

# What's next?

- ▶ Image Processing Techniques.
- ▶ Video Processing.
- ▶ Demonstration of some Computer Vision applications.

**Note- The best is yet to come. 😊**

# Thank-you

Ekansh Gupta, Part 3, ECE - **7054223124**

Parth Shyara, Part 3, ECE - **7068254257**



<https://www.facebook.com/robotics.iitbhu>