Introduction to MATLAB

Mohamad Nassereddine

Subject Outline

- Discuss the subject outline in details

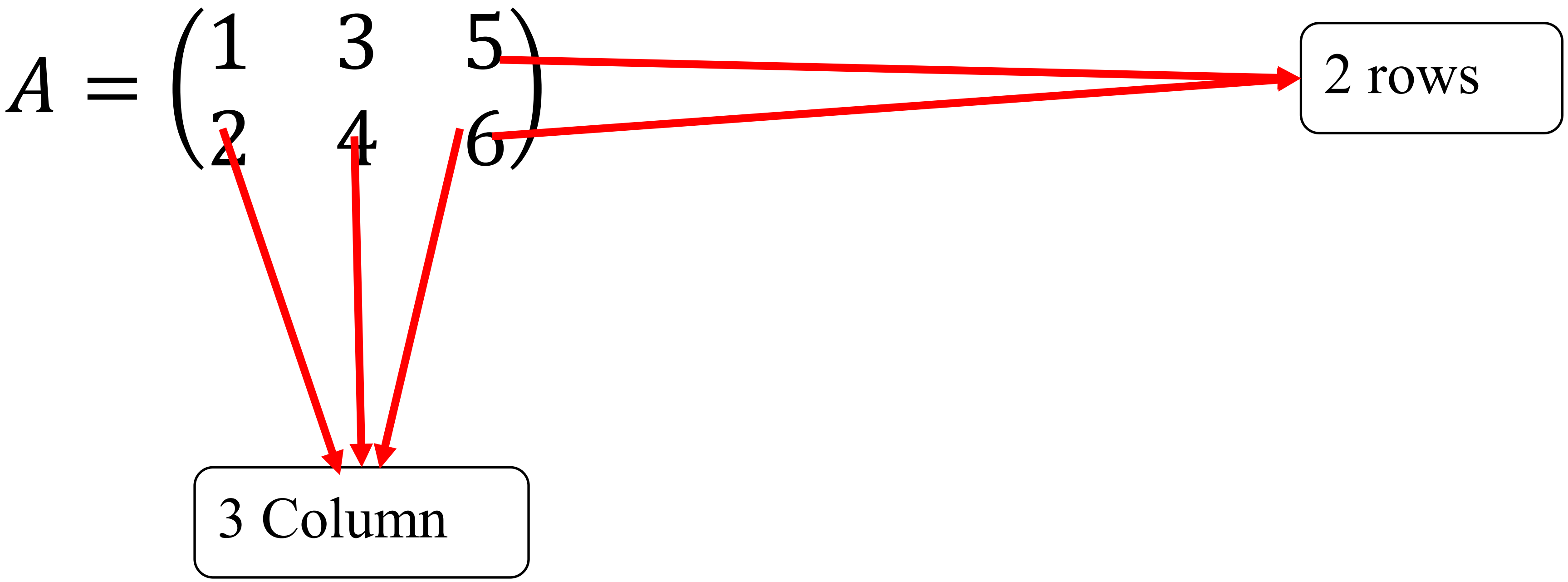
Objectives

- To enable you to use some simple MATLAB commands from the Command Window.
- To examine various MATLAB desktop and editing features.
- To learn some of the new features of the MATLAB R2018b Desktop.
- To learn to write scripts in the Editor and Run them from the Editor.
- To learn some of the new features associated with the tabs (in particular, the PUBLISH and APPS features).

Introduction

- MATLAB is a powerful technical computing system for handling scientific and engineering calculations. The name MATLAB stands for Matrix Laboratory, because the system was designed to make matrix computations particularly easy.
- A matrix is an array of numbers organized in *m* rows and *n* columns. An example is the following $m \times n = 2\times3$ array:
- $A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$

Introduction



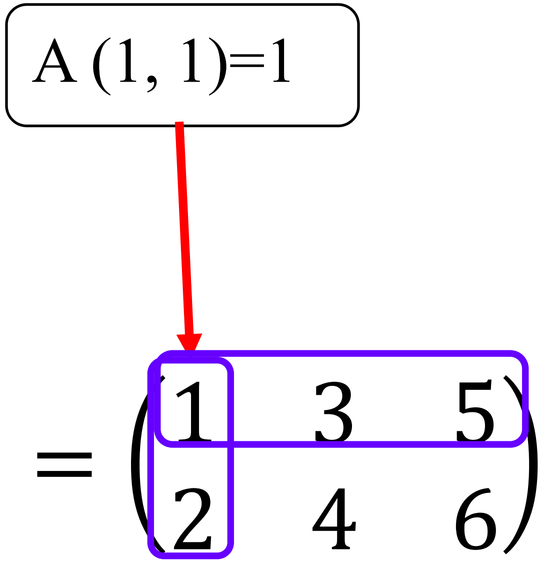
- $A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$

2 rows

3 Column

Introduction

- How to identify the location?



A (1, 1)=1

- $A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$

A (1, 2)=3

A (2, 2)=?

A (1, 3)=?

A (2, 3)=?

A (2, 1)=?

Introduction

- There are three essential requirements for successful MATLAB applications:
  - You must learn the ***exact* rules** for writing MATLAB statements and using MATLAB utilities.
  - You must know the mathematics associated with the problem you want to solve.
  - You must develop a logical plan of attack—the algorithm—for solving a particular problem.

Introduction

- This chapter is devoted mainly to the first requirement: learning some basic MATLAB rules.
- Computer programming is the process of writing a collection of instructions that performs a specific task when executed by a computer.
- In this book you are introduced to programming by using some of the capabilities of MATLAB to do technical computing.

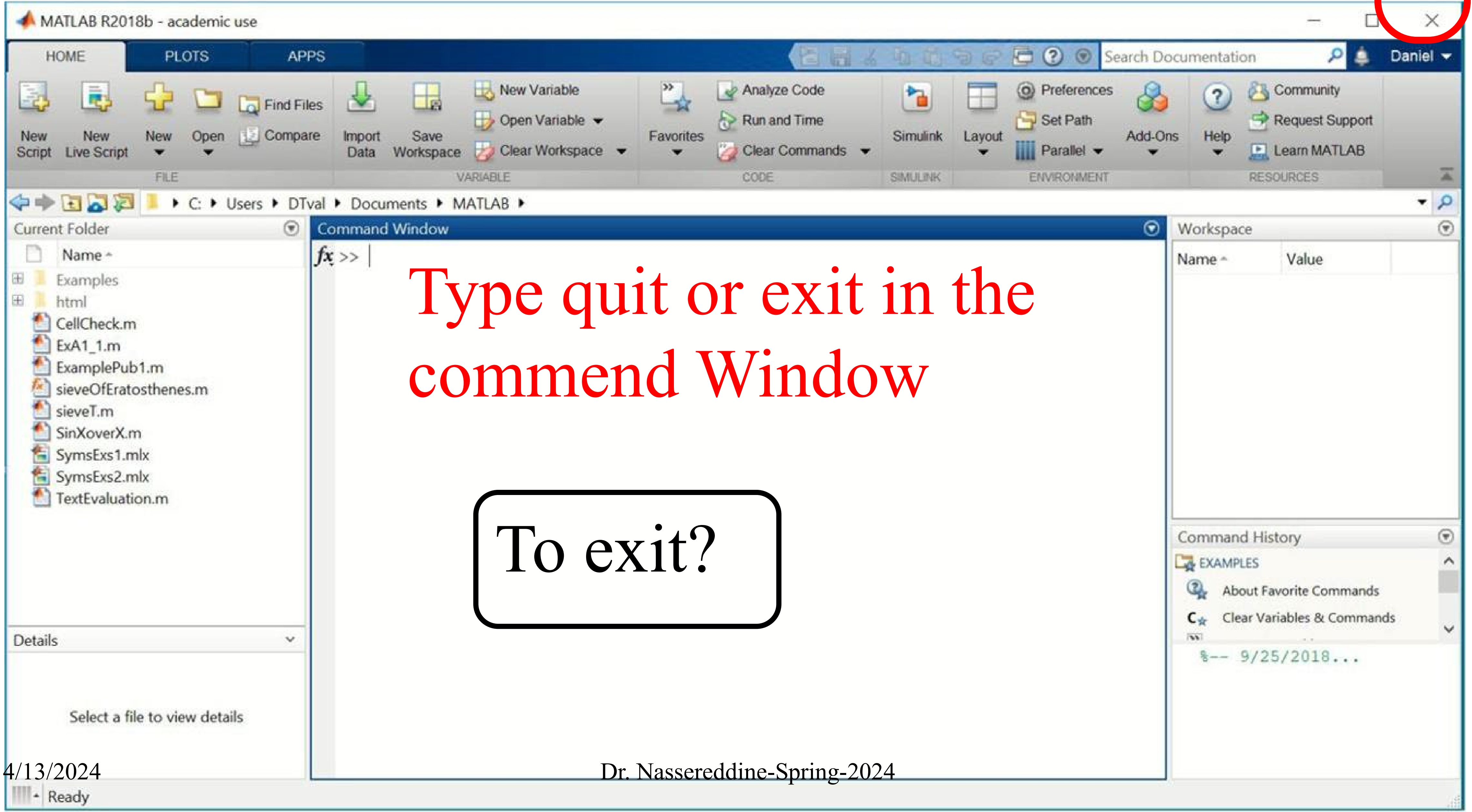
Introduction

- Important
- Don’t be concerned about understanding exactly what is happening. Understanding will come with the work you need to do in later chapters.
- It is very important for you to practice with MATLAB to learn how it works. Once you have grasped the basic rules in the first chapter, you will be prepared to master many of those presented in the next chapter and in the Help files provided with MATLAB.
- This will help you go on to solve more interesting and substantial problems. In the last section of this chapter you will take a quick tour of the MATLAB desktop.

MATLAB

- Either MATLAB must be installed on your computer or you must have access to a network where it is available. Throughout this book the latest version at the time of writing is assumed (version R2018b).
- To start from Windows, double-click the MATLAB icon on your Windows desktop.
- To start from UNIX, type matlab at the operating system prompt. To start from MAC-OS open X11 (i.e., open an X-terminal window), then type matlab at the prompt. The MATLAB desktop opens

MATLAB

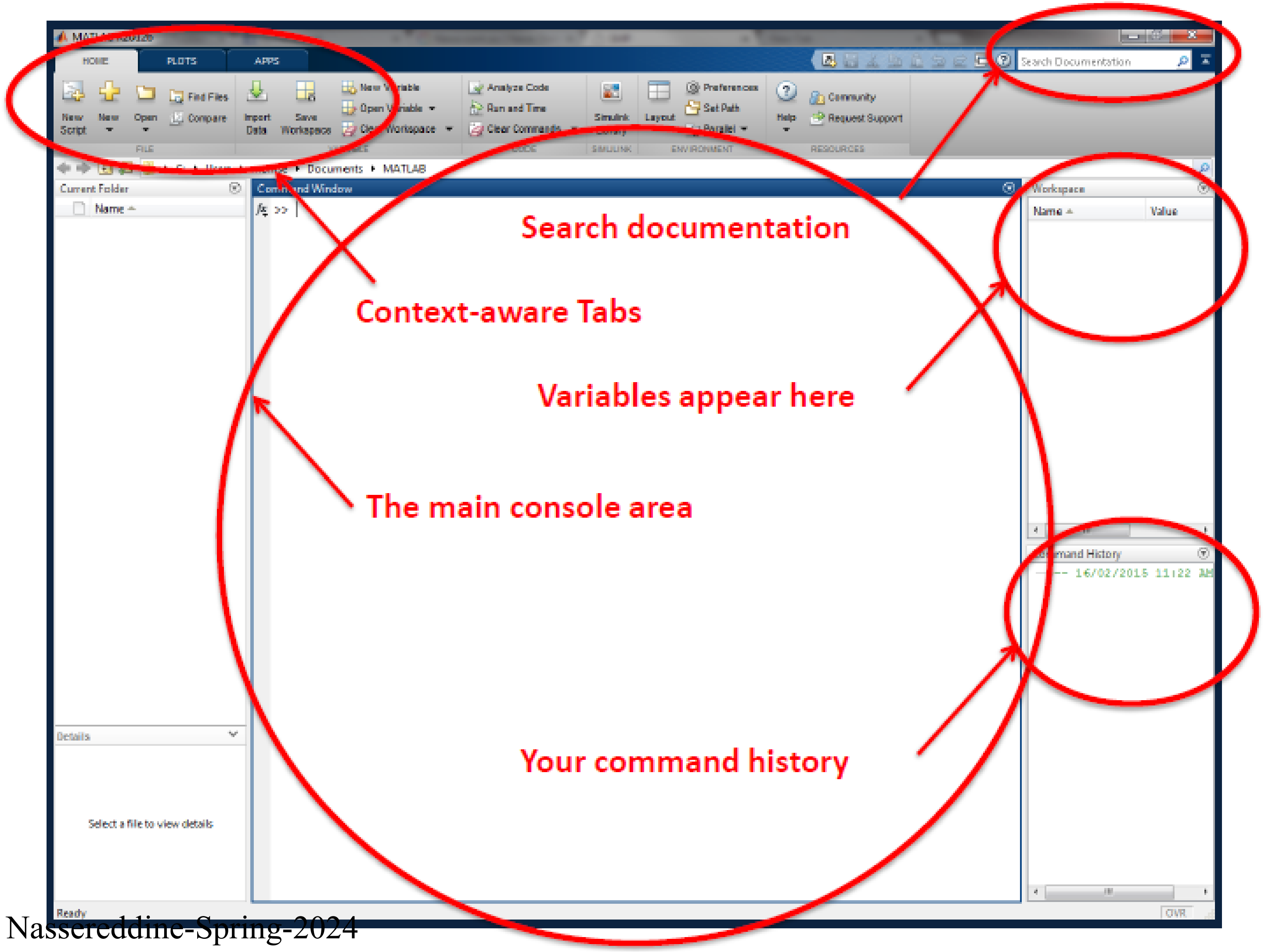


Type quit or exit in the commend Window

To exit?

MATLAB

- Matlab window explanation



Search documentation

Context-aware Tabs

Variables appear here

The main console area

Your command history

MATLAB- Functions & Commends

- MATLAB comes with a large number of in-built basic commands that allow you to start working with the program environment immediately.
  - Command for managing sessions
  - Commands for working with the fie system
  - Input/output commands
  - Mathematic functions
  - Plotting functions
  - And many more

MATLAB- Functions & Commends

- MATLAB has numerous general functions. Try date and calendar for starters.
- It also has numerous *commands*, such as clc (for *clear command window*). Help is one you will use a lot
- The difference between functions and commands is that functions usually return with a value (e.g., the date), while commands tend to change the environment in some way (e.g., clearing the screen or saving some statements to the workspace).

MATLAB- Functions & Commends

Command	Purpose
clc	Clears command window
clear	Removes variables from memory
exist	Checks for existing of file or variable
help	Searches for a help topic
lookfor	Searches help entries for a keyword
quit	Stops matlab
who	List current variables

MATLAB- Functions & Commends

Command	Purpose
whos	Lists current variables (long display)
disp	Displays contents of an array or string
fscanf	Read formatted data from a file
format	Controls screen-display format
fprintf	Performs formatted writes to screen or file
input	Displays prompts and waits for input
;	Suppresses screen printing

MATLAB

- Starting MATLAB automatically creates a folder named MATLAB in the user’s Documents Folder.
- This feature is quite convenient because it is the default working folder. It is in this folder that anything saved from the Command Window will be saved.
- Now you can experiment with MATLAB in the Command Window. If necessary, make the Command Window active by placing the cursor in the Command Window and left-clicking the mouse button anywhere inside its border.

MATLAB- Arithmetic

- Type 2+3 after the >> prompt, followed by **Enter** (press the **Enter** key) as indicated by <**Enter**>:
- >> 2+3 <**Enter**>

- The sympols:
  - * mean multiply
  - / mean divide
  - ^ mean exponentiation
  - \ mean the denominator is to the left of the simple and the number is to the right

MATLAB- Arithmetic

- Here are hints on creating and editing command lines:
  - The line with the >> prompt is called the *command line*.
  - You can edit a MATLAB command before pressing **Enter** by using various combinations of the **Backspace**, **Left-arrow**, **Right-arrow**, and **Del** keys. This helpful feature is called *command-line editing*.
  - You can select (and edit) commands you have entered using **Up-arrow** and **Down-arrow**. Remember to press **Enter** to have the command carried out (i.e., to *run* or to *execute* the command).
  - MATLAB has a useful editing feature called *smart recall*. Just type the first few characters of the command you want to recall. For example, type the characters 2* and press the **Up-arrow** key—this recalls the most recent command starting with 2*.

MATLAB- Arithmetic

- How do you think MATLAB would handle 0/1 and 1/0?

- If you insist on using ∞ in a calculation, which you may legitimately wish to do, type the symbol Inf (short for *infinity*). Try 13+Inf and 29/Inf.

- Another special value that you may meet is NaN, which stands for *Not-a- Number*. It is the answer to calculations like 0/0.

MATLAB

- Variables
- A variable does not mean the same thing in programming as it does in maths
- A variable in programming can be thought of as a container with a label. We can store data in the container and then use the variable name to retrieve the data later
- The variable name can be anything alpha-numerical and must start with a letter not a number
- A_4 is acceptable, 4_A is not acceptable

MATLAB- Arithmetic

- Under numerous analysis, it is important to assign values to a variable.
- In MATLAB, we can assign variable as follows:
  - First enter the command (*statement* in programming jargon) a = 2.
  - The MATLAB command line should look like this:
  - >> a = 2 <**Enter**>

MATLAB- Arithmetic

- Now enter the statement
- `>> b = 3;` <**Enter**>
- The semicolon (;) prevents the value of b from being displayed. However, b still has the value 3,
- `>> x = 2; y = 3;` <**Enter**>
- `>> z = x + y` <**Enter**>
- Notice that, in addition to doing the arithmetic with variables with assigned values, several commands separated by semicolons (or commas) can be put on one line.

MATLAB- Arithmetic

- >> x = 2; y = 3; <**Enter**>
- >> z = x + y <**Enter**>
- *The variable can be overwritten*
- >> z=z+5 <Enter>
- Notice that, the new value for z that will be stored now is 10
- To erase, type:

  clear z;

MATLAB- Mathematical Function

- It is important to get familiar with MATLAB Mathematical Functions.
  - Sqrt(2) is the square root of the number 2
  - Sqrt(pi)
  - Sin(x). It is worth noting that x is in radians and not degrees
  - To convert from degrees to radians
  - $rad = degree \times \frac{\pi}{180}$
  - Exp(x) is the exponential function in MATLAB

MATLAB- Mathematical Function

- Because of the numerous built-in functions like pi or sin, care must be taken in the naming of user-defined variables.
- Names should not duplicate those of built-in functions without good reason.

MATLAB

- MATLAB defines various data:
  - Data types:
    - Integer (memory efficient)
    - Double (store real numbers)
    - Character (store text)
    - Logical (true/false)
    - Cell arrays
    - Matrices
    - Tables

MATLAB

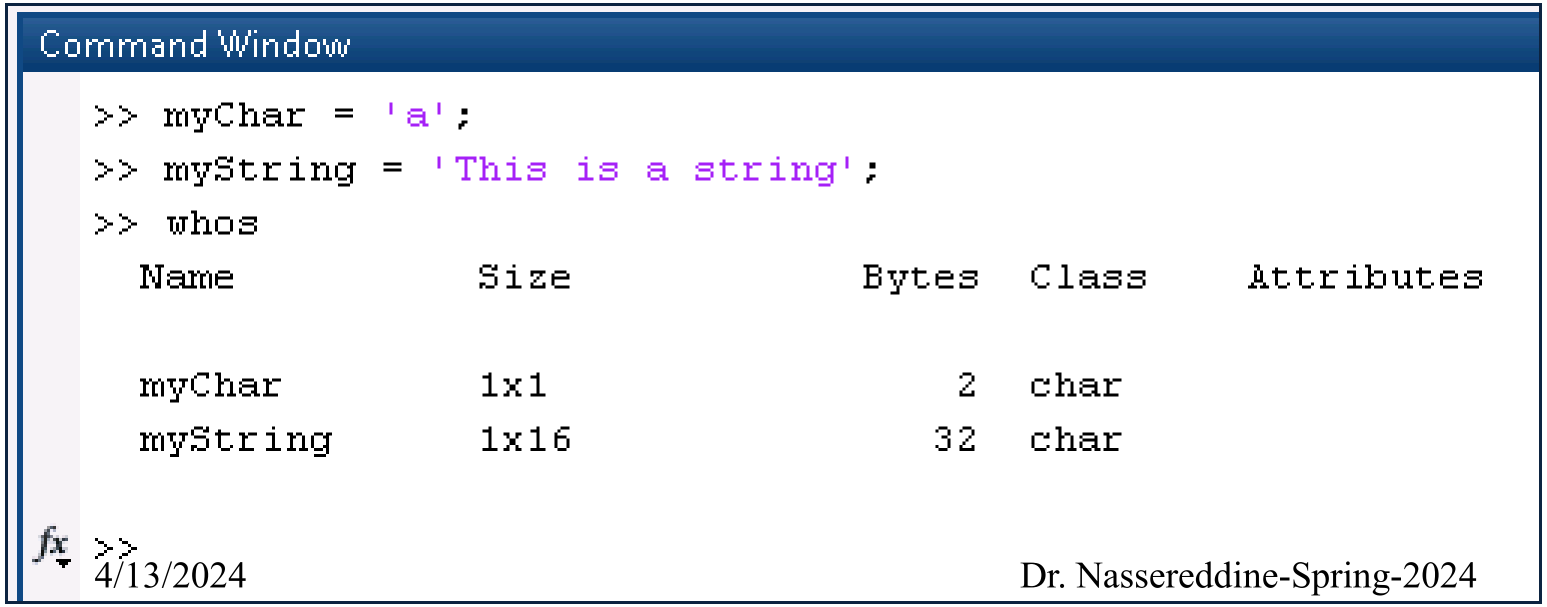
- In MATLAB, Data isn’t only numeric, it can be test or logic:
- Example:

```
client(1). Name=‘Alfred Alfred’
client(1).billing = 540
client(1).result =
```

MATLAB

- Characters
- Characters are actually stored as number as determined by the ASCII code chart

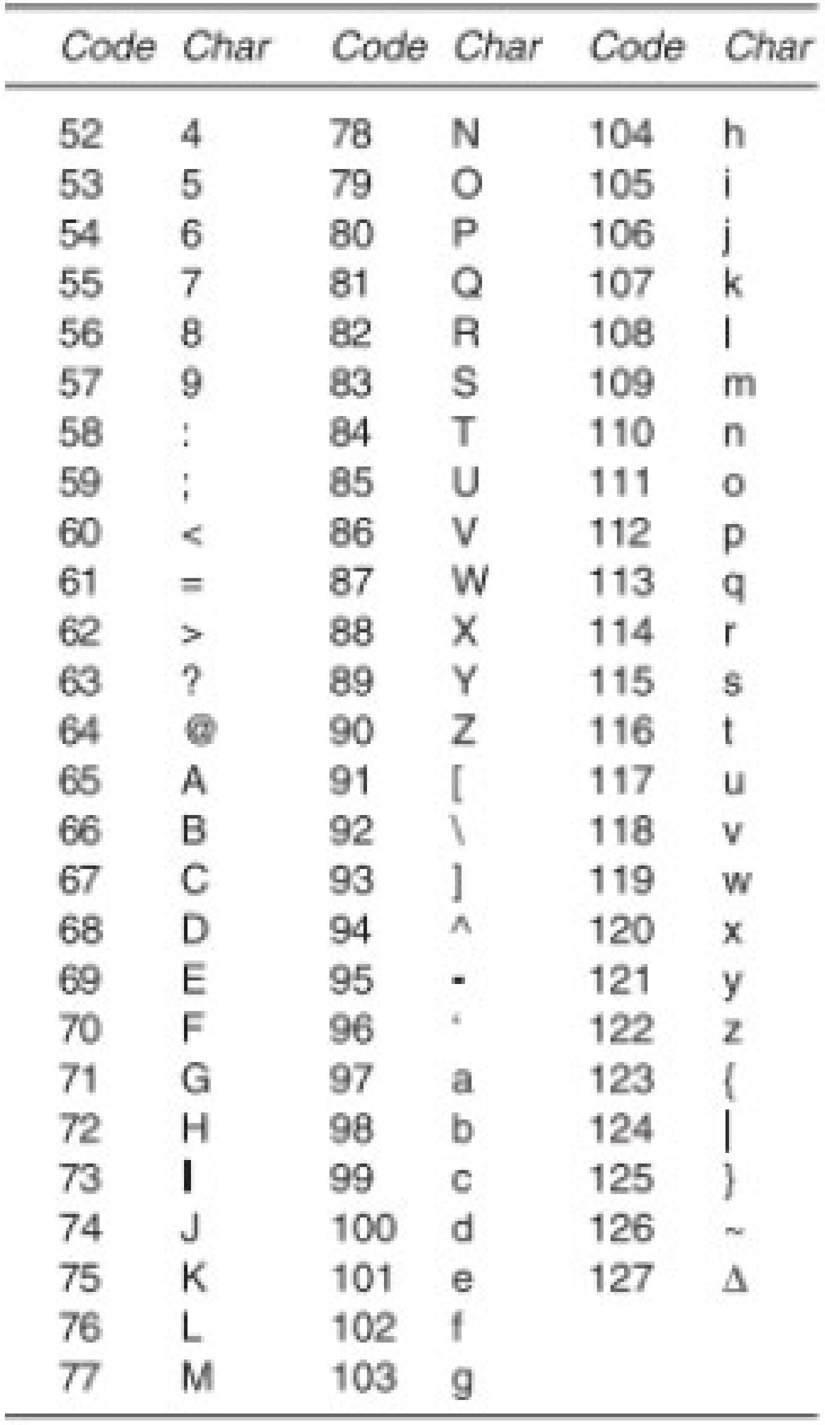
American Standard Code for Information Interchange



```
>> myChar = 'a';
>> myString = 'This is a string';
>> whos
  Name            Size            Bytes  Class     Attributes

  myChar          1x1                 2  char
  myString        1x16               32  char

>>
```



Code	Char	Code	Char	Code	Char
52	4	78	N	104	h
53	5	79	O	105	i
54	6	80	P	106	j
55	7	81	Q	107	k
56	8	82	R	108	l
57	9	83	S	109	m
58	:	84	T	110	n
59	;	85	U	111	o
60	<	86	V	112	p
61	=	87	W	113	q
62	>	88	X	114	r
63	?	89	Y	115	s
64	@	90	Z	116	t
65	A	91	[	117	u
66	B	92	\	118	v
67	C	93	]	119	w
68	D	94	^	120	x
69	E	95	_	121	y
70	F	96	`	122	z
71	G	97	a	123	{
72	H	98	b	124	|
73	I	99	c	125	}
74	J	100	d	126	~
75	K	101	e	127	Δ
76	L	102	f		
77	M	103	g		

MATLAB

- To create a string, the string text must be surrounded by apostrophes
- Otherwise MATLAB will think you have written is variable or function)
- For example:

```
X = 3;
Y = “X’;
Z = X;
```

MATLAB

- Numbers can also be stored as Characters

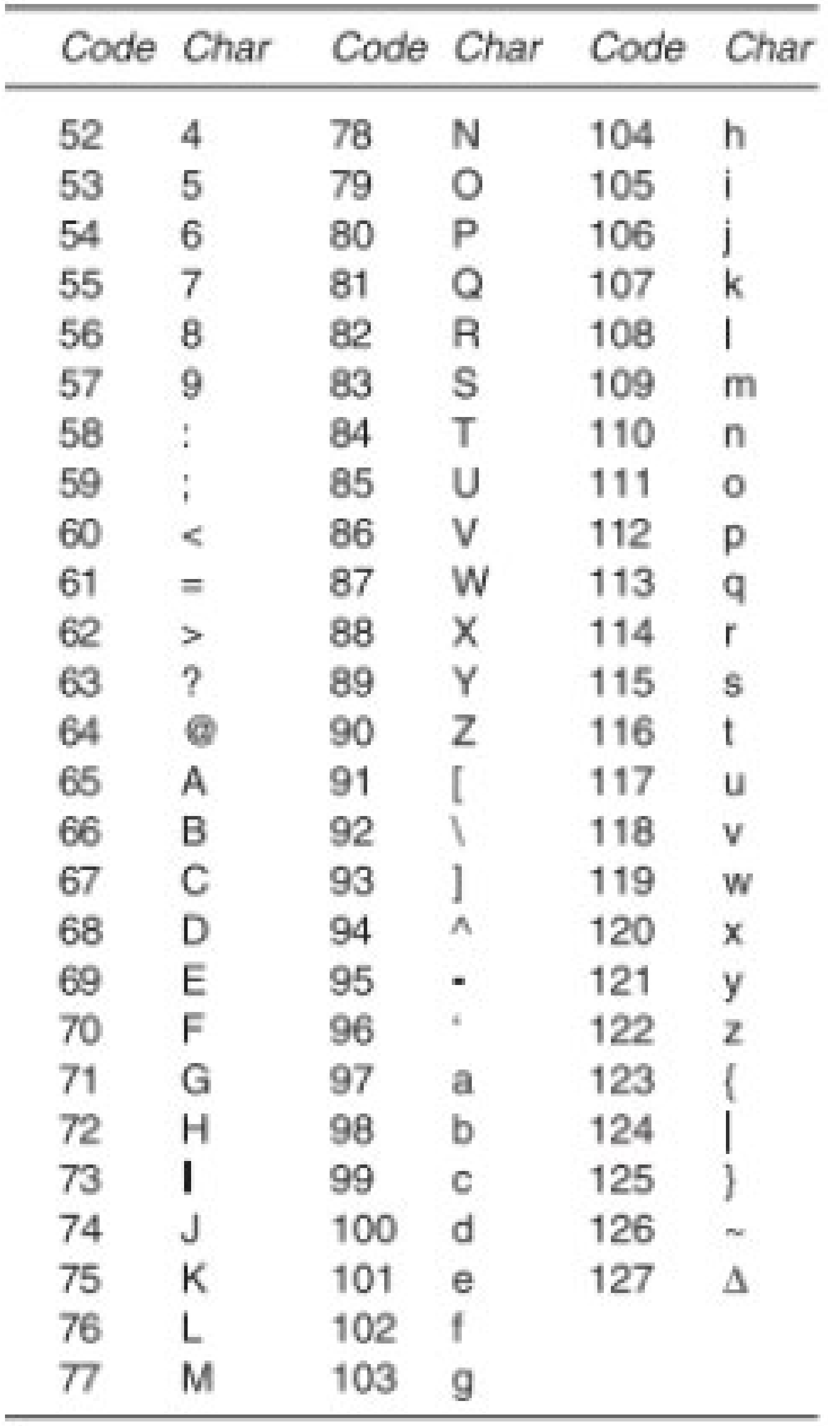
X = ‘4’

Y = X+1;

the number 53 will be stored in Y

To convert from a double to char according to the table, you can use:

Y = char (y) ;



Code	Char	Code	Char	Code	Char
52	4	78	N	104	h
53	5	79	O	105	i
54	6	80	P	106	j
55	7	81	Q	107	k
56	8	82	R	108	l
57	9	83	S	109	m
58	:	84	T	110	n
59	;	85	U	111	o
60	<	86	V	112	p
61	=	87	W	113	q
62	>	88	X	114	r
63	?	89	Y	115	s
64	@	90	Z	116	t
65	A	91	[	117	u
66	B	92	\	118	v
67	C	93	]	119	w
68	D	94	^	120	x
69	E	95	-	121	y
70	F	96	‘	122	z
71	G	97	a	123	{
72	H	98	b	124	|
73	I	99	c	125	}
74	J	100	d	126	~
75	K	101	e	127	Δ
76	L	102	f		
77	M	103	g		

MATLAB

- If you have a double or integer that you want to be expressed in a string form (or vice versa) you can use the functions **num2str** or **str2num**.
- A function is called by typing the name of the function, followed by parentheses () containing any inputs separated by commas. If a function provides an output, that output can be stored in a variable using an equals sign as normal.

```
X=5;
Y=num2str(X);
Z=str2num(Y);
```

- After executing this code, X and Z will both contain the number 5 in double form. Y will contain '5' in string form.

MATLAB

- Say I have stored a solution to a problem in variable **sol**, which is of type double. I then want to create a string that says ‘The answer is:’ followed by the solution stored in **sol**.
- To create this sentence, I need to convert the value stored in **sol** to a string type and then concatenate (put together) the two strings.
- The concatenate anything in MATLAB, square brackets are used [], with each element separated by commas.

```
sol=10;
[‘The answer is: ’, num2str(sol)]
```

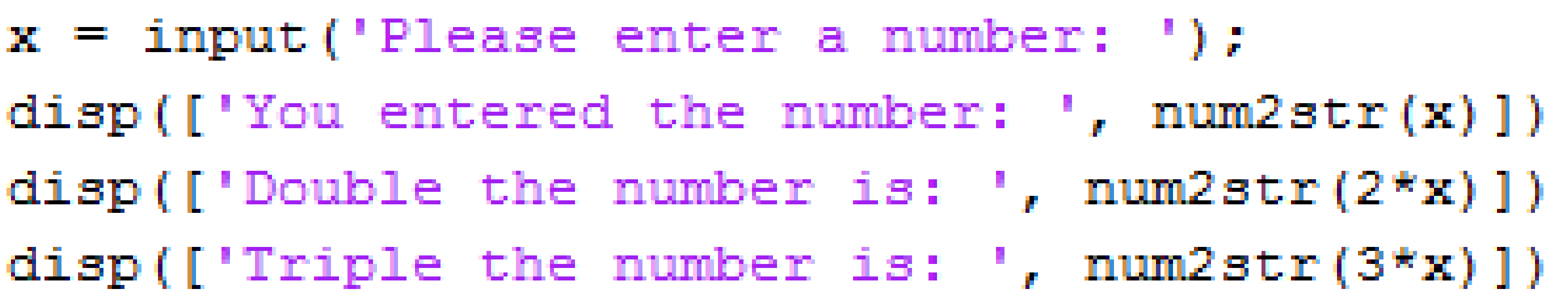
MATLAB

- A string can be displayed using the function **disp**.
- Remember, a function is called by typing the name of the function, followed by parentheses () containing any inputs separated by commas.
- If a function provides an output, that output can be stored in a variable using an equals sign as normal.
- However, **disp** has no output, and just one input which is the string you want to display to the user.

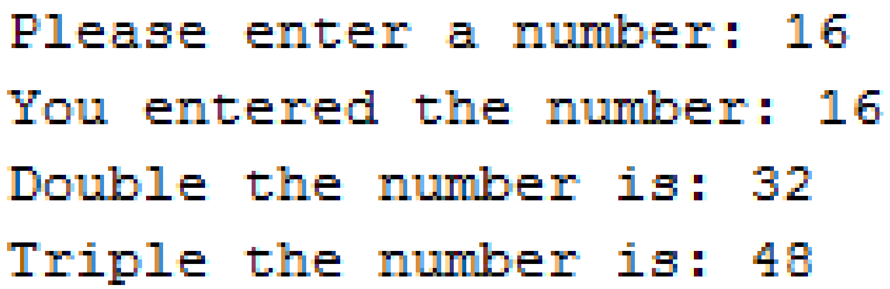
Disp (‘I’’m learning to program!’)

MATLAB

- To interact with the user, we can use the function **input**.
- The function **input** has just one input, which is a string to be displayed to the user before they enter their information. The output is a double of whatever the user typed before pressing enter.



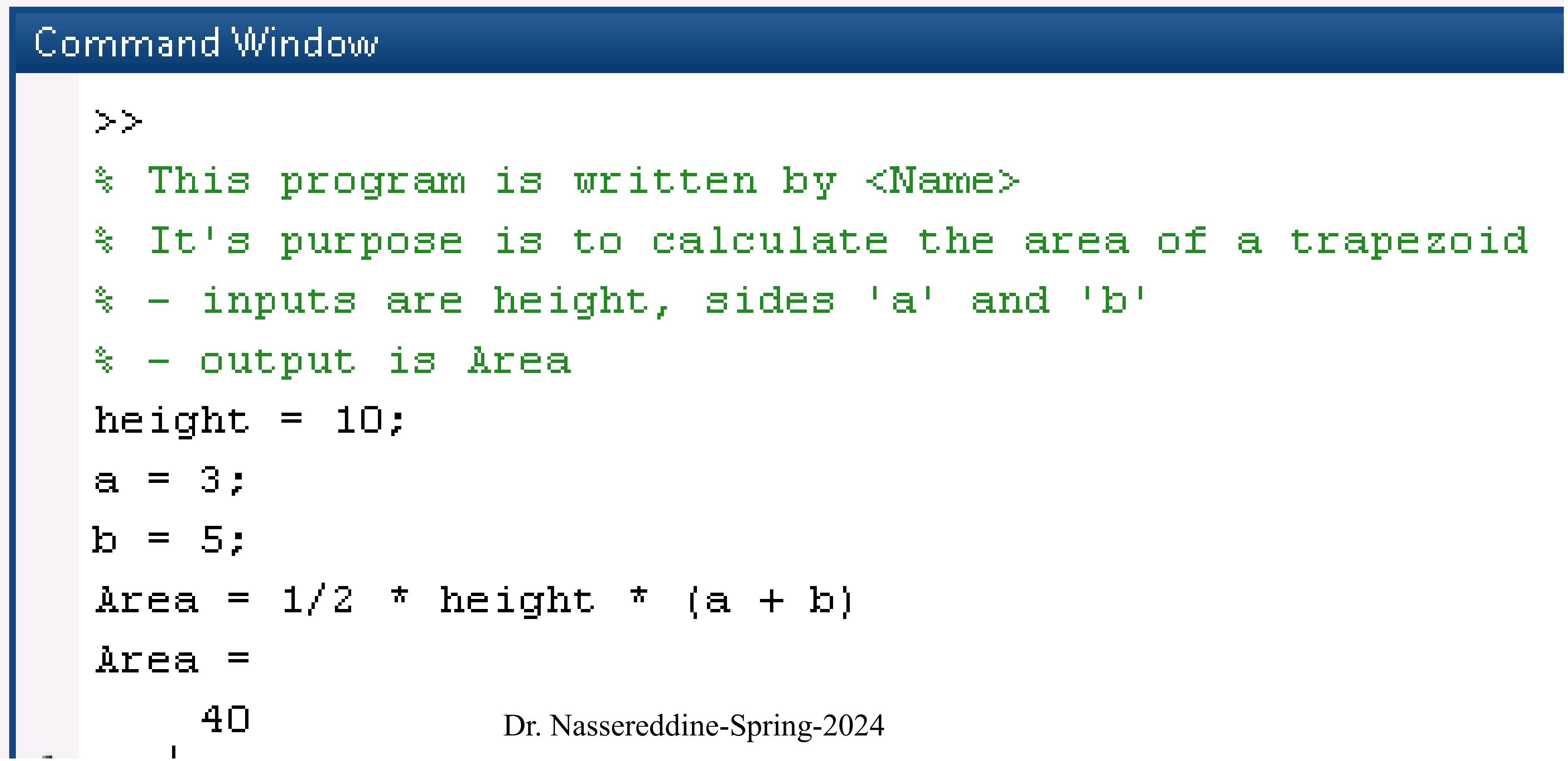
```
x = input('Please enter a number: ');
disp(['You entered the number: ', num2str(x)])
disp(['Double the number is: ', num2str(2*x)])
disp(['Triple the number is: ', num2str(3*x)])
```



```
Please enter a number: 16
You entered the number: 16
Double the number is: 32
Triple the number is: 48
```

MATLAB

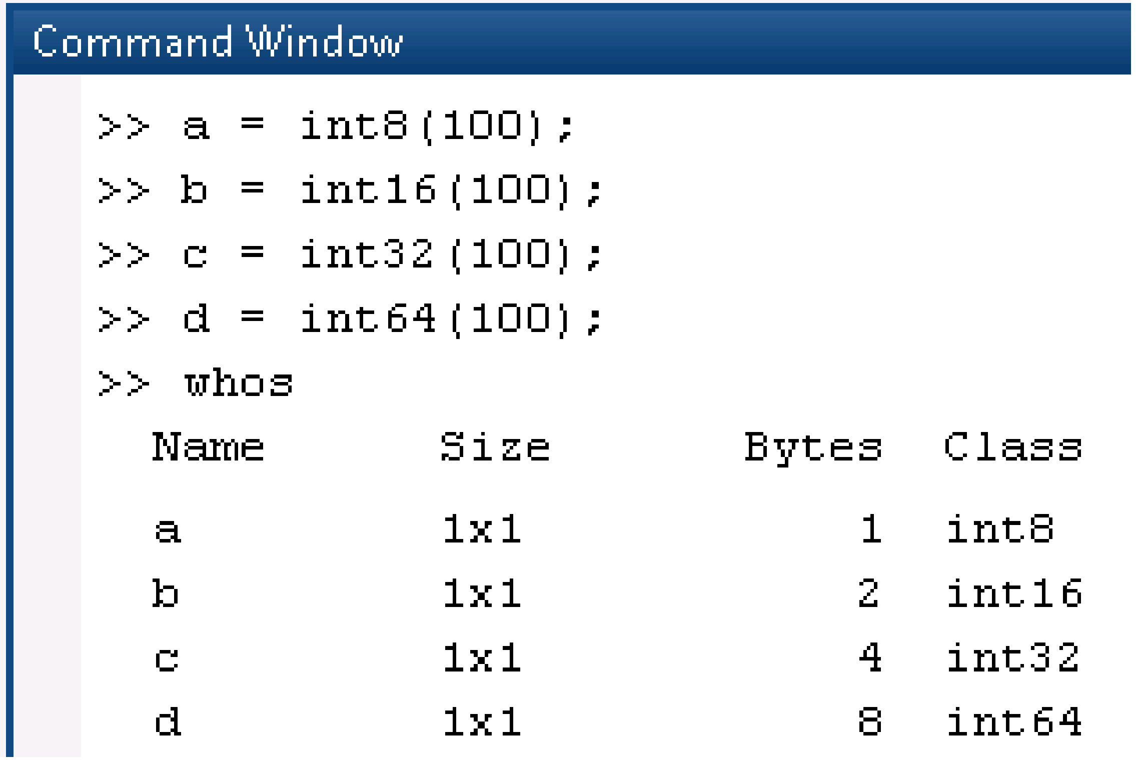
- Commenting is an excellent way to keep track of the actions your program is performing. It also helps other people read your program and gain a better understanding of what you are trying to accomplish. •Commenting will become more important in later, in more complicated assignments



```
>>
% This program is written by <Name>
% It's purpose is to calculate the area of a trapezoid
% - inputs are height, sides 'a' and 'b'
% - output is Area
height = 10;
a = 3;
b = 5;
Area = 1/2 * height * (a + b)
Area =
    40
```

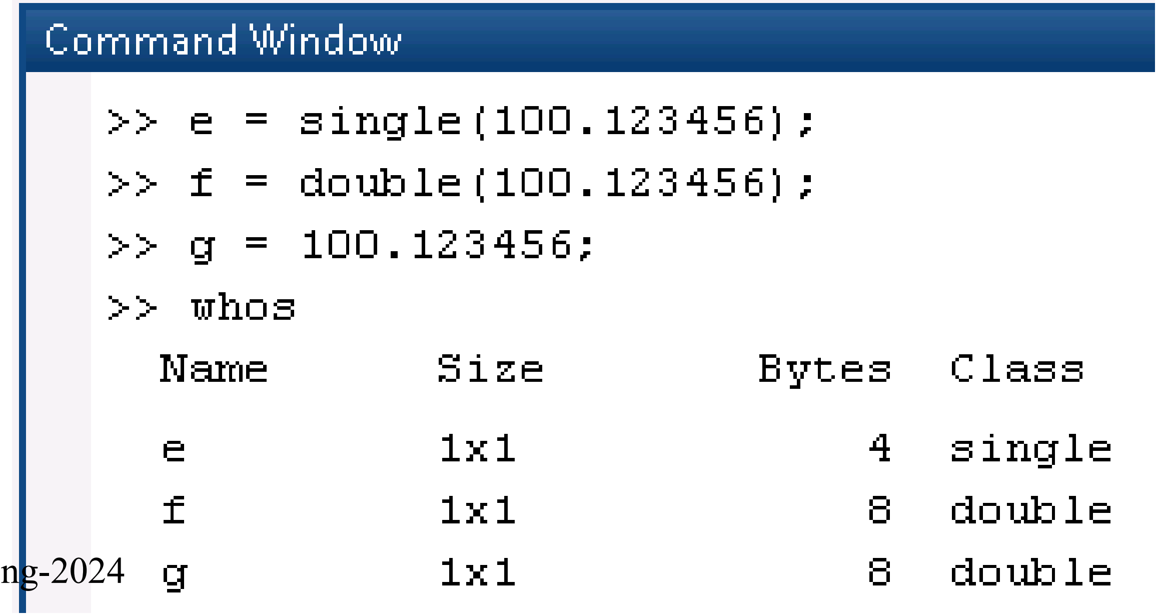
MATLAB

- **Integers** Can’t store decimals, only whole numbers
- 8-, 16-, 32-and 64-bit integers
- **Floating Point** Can store decimals
- Takes up more space!
  - Largest numbers possible : Single: 3.4028×10^{38}
  - Double: 1.7977×10^{308}



```
>> a = int8(100);
>> b = int16(100);
>> c = int32(100);
>> d = int64(100);
>> whos
  Name          Size              Bytes  Class

  a             1x1                   1  int8
  b             1x1                   2  int16
  c             1x1                   4  int32
  d             1x1                   8  int64
```

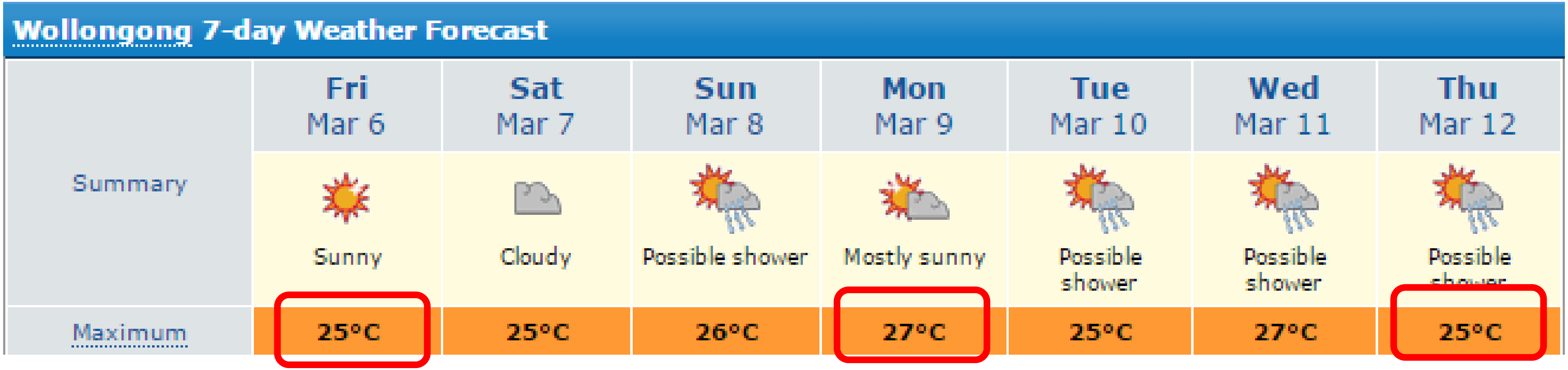


```
>> e = single(100.123456);
>> f = double(100.123456);
>> g = 100.123456;
>> whos
  Name          Size              Bytes  Class

  e             1x1                   4  single
  f             1x1                   8  double
  g             1x1                   8  double
```

MATLAB

- Example of an array



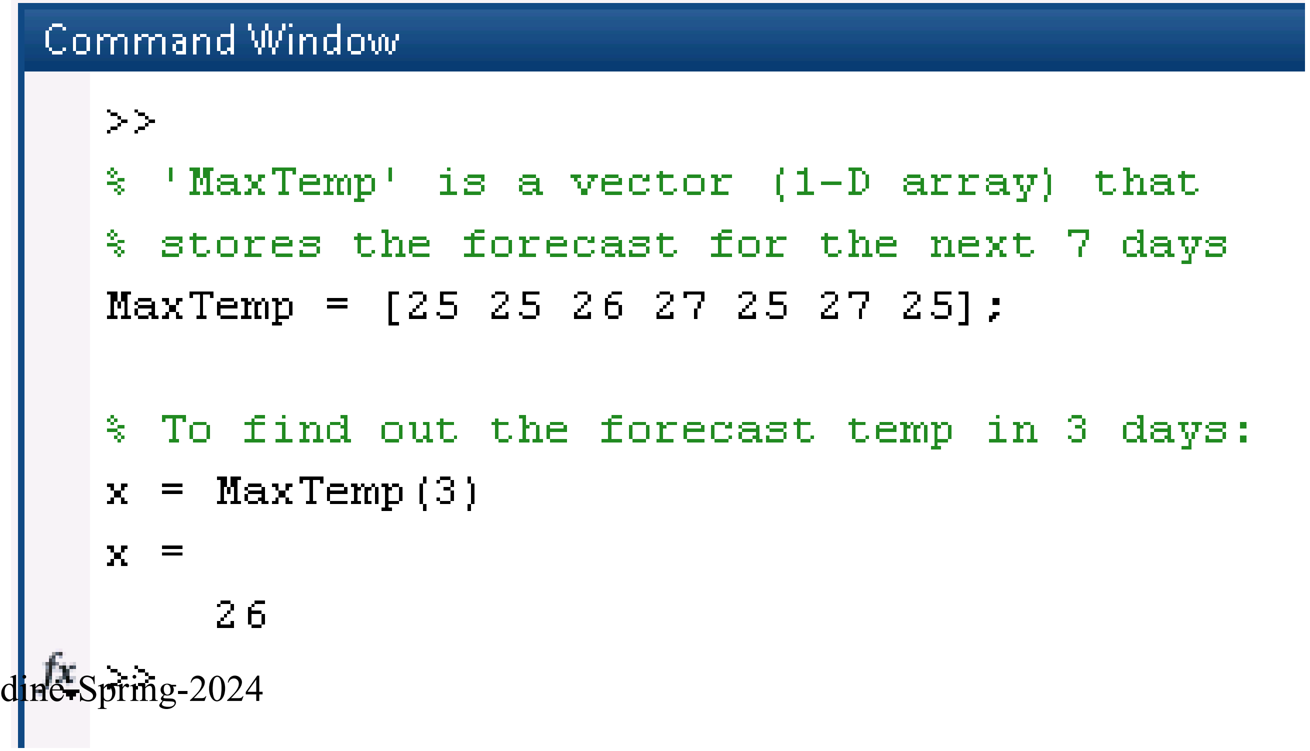
Wollongong 7-day Weather Forecast

	Fri Mar 6	Sat Mar 7	Sun Mar 8	Mon Mar 9	Tue Mar 10	Wed Mar 11	Thu Mar 12
Summary	Sunny	Cloudy	Possible shower	Mostly sunny	Possible shower	Possible shower	Possible shower
Maximum	25°C	25°C	26°C	27°C	25°C	27°C	25°C

MATLAB

- Array and vectors
- We can create an array for maximum temperature throughout the week called ‘**MaxTemp**’ This structure can be said to hold 1-D data(one dimension)
- In MATLAB these are called ***vectors***. However, in other programming languages, they might simply use ‘array’.

The number we use to access the correct element in the array is known as the ***index*** and starts at 1



```
>>
% 'MaxTemp' is a vector (1-D array) that
% stores the forecast for the next 7 days
MaxTemp = [25 25 26 27 25 27 25];

% To find out the forecast temp in 3 days:
x = MaxTemp(3)
x =
    26
fx >>
```

MATLAB

- Changing vectors
- If I have created an array:

  X=[1, 3, 5, 8];

- And I want to change the 4th element of X to a 7, I would type:

  X(4)=7;

- As opposed to:

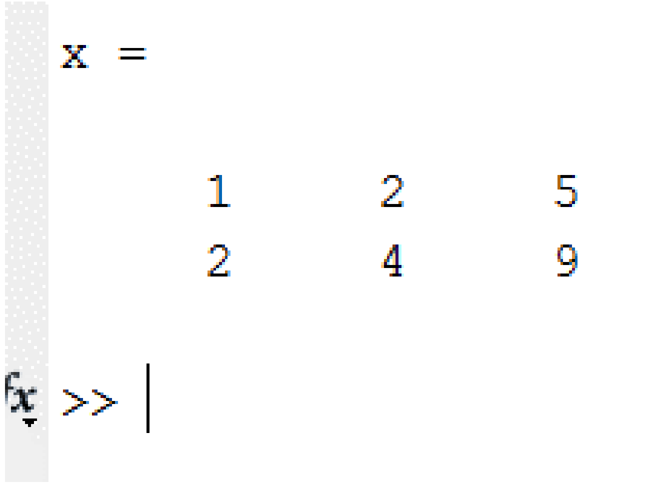
  X=7;

- Where the entire array is deleted and replaced with 7.

MATLAB

- I can create a 2-D array by separating terms by commas (spaces also work) and using a semicolon to represent a new row:

  X=[1, 2, 5; 2, 4, 9];



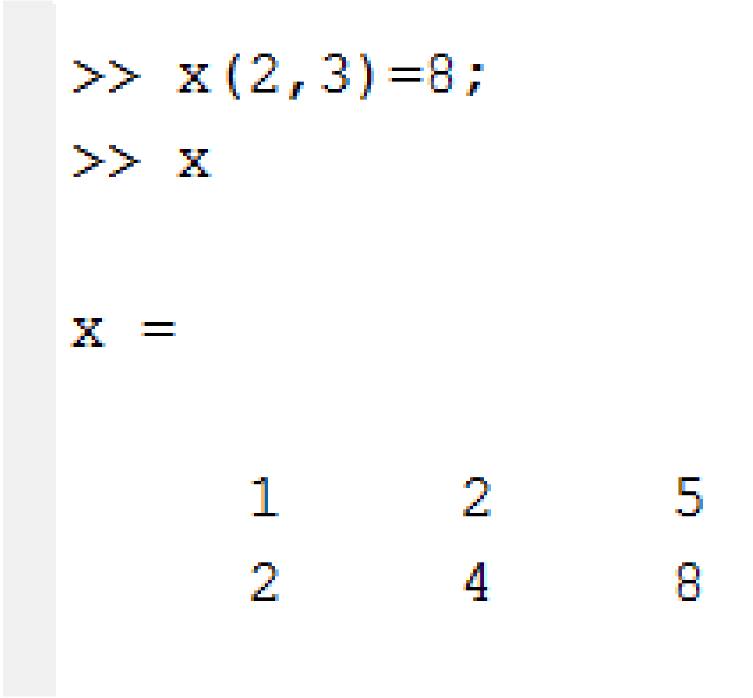
```
x =

     1     2     5
     2     4     9

>> 
```

- If I want to change the element in row 2, column 3 of X to an 8, I would type:

  X(2,3)=8;



```
>> x(2,3)=8;
>> x

x =

     1     2     5
     2     4     8
```

MATLAB

- Cell arrays can contain different types data in each cell.
- If all elements are all numeric, or you are concatenating strings, use square brackets [ ] to create arrays and parentheses ( ) to access elements in the array.
- If you are combining different types of data, or you want individual strings in individual elements in an array, use braces { } to create and access elements from arrays.

```
>> w=[1,3,7];
x=['Combining', ' Strings'];
y={'Mixed', 2};
z={'Individual', 'Strings'};
>> w(2)
ans= 3
>> x(2)
ans= ‘o’
>> y{2}
ans= 2
>> z{2}
ans= Strings
>> z{2}(2)
ans=t
```