ME 444
MATLAB® FOR ENGINEERS

Lecturer:

Dr. Nurettin Furkan DOĞAN

Mechanical Engineering Department

4 th Floor Office No: 303

nfdogan@gantep.edu.tr

0342 317 1200- 2569

Office Hours:

Tuesday: 13.30- 15:00

Friday : 10.00- 11.30

CHAPTER 3

MATLAB FUNDAMENTALS

Main features of MATLAB

- MATLAB is a high-performance application software and programming language named after the words MATrix LAboratory.
- The underlying structure of MATLAB is matrices that do not require dimensioning.
- That is, all the inputs and outputs we do define a matrix without requiring a command.
- First written in Fortran language, MATLAB was later written in C language.

Main features of MATLAB

- Easy and effective programming in a high-level language,
- Having an interactive interface for rapid development
- Vectorized calculations for efficient programming and automatic memory allocation.
- Built-in support for numerical calculation methods.
- Have a variety of modern data structures and types, including complex numbers.
- High quality graphics and visualization.
- Symbolic math toolbox for algebraic operations and differential equations

Main features of MATLAB

- It can be used directly through the Command Window or users can design and use their own programs.
- Simulation support with SIMULINK
- Cross-platform portable program files
- Numerous add-on toolboxes for applications and simulations
- User-contributed database of files and toolboxes, including numerous lectures and demos
- Plugin support for other programming languages such as C/C++, Java.

Main features of MATLAB

➢ Optional toolboxes

- Toolboxes are collections that contain functions developed for special applications. Among these applications:

✓ Symbolic Operations

✓ Image Processing

✓ Statistics

✓ Control Systems Design

✓ Artificial Neural Networks (Deep Learning Toolbox)

Main features of MATLAB

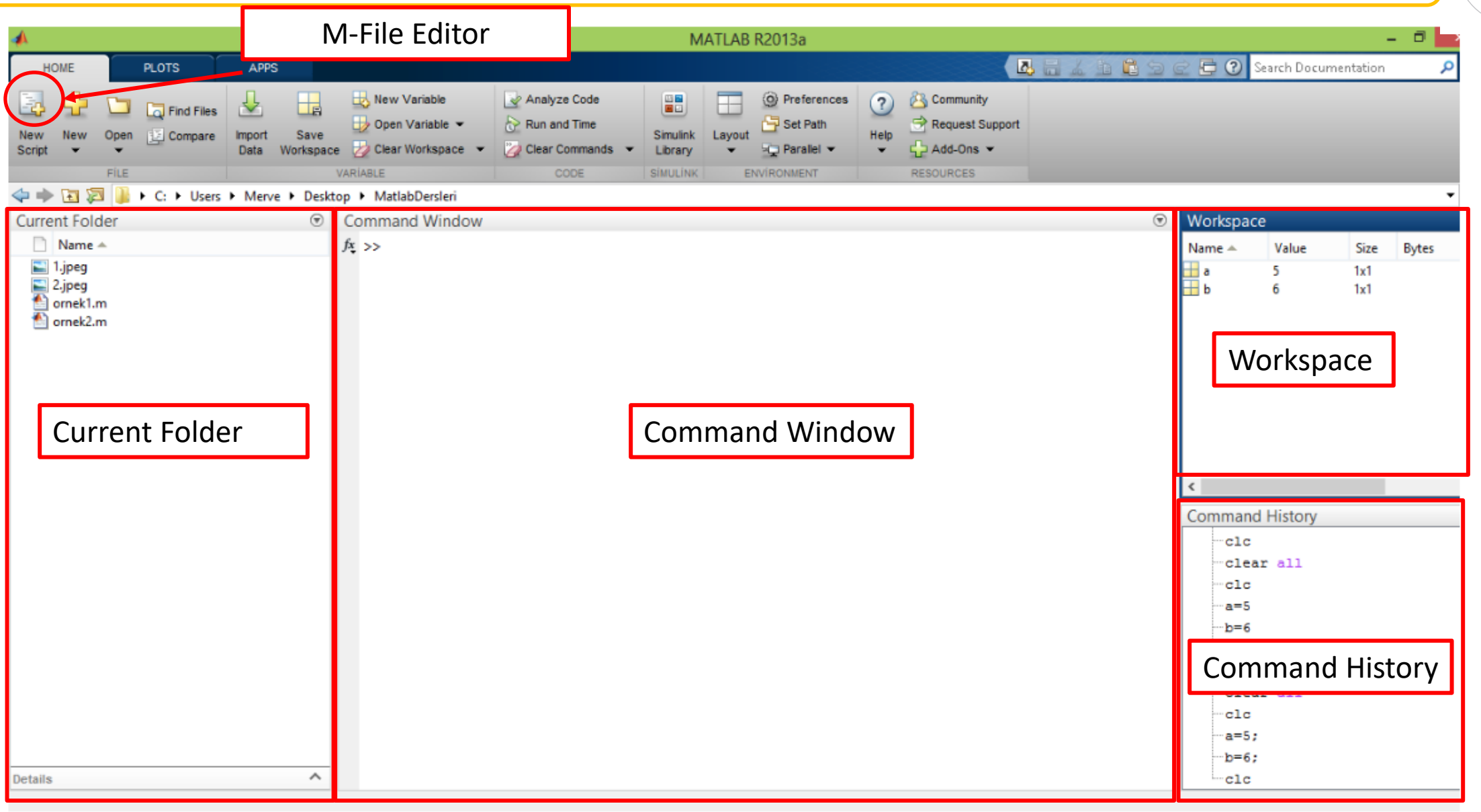
- There are built-in programs in MATLAB. These programs are called *functions*.
- The use of MATLAB functions is identical to the use of the *y=f(x)* function in mathematics.
- For example, in the a=sin(x) function, the *sin* function calculates *the sine of the angle x* (input-input value);
- The user assigns this value to a variable **a**.
- The value **a** is an *output of the sin function*.

Main features of MATLAB

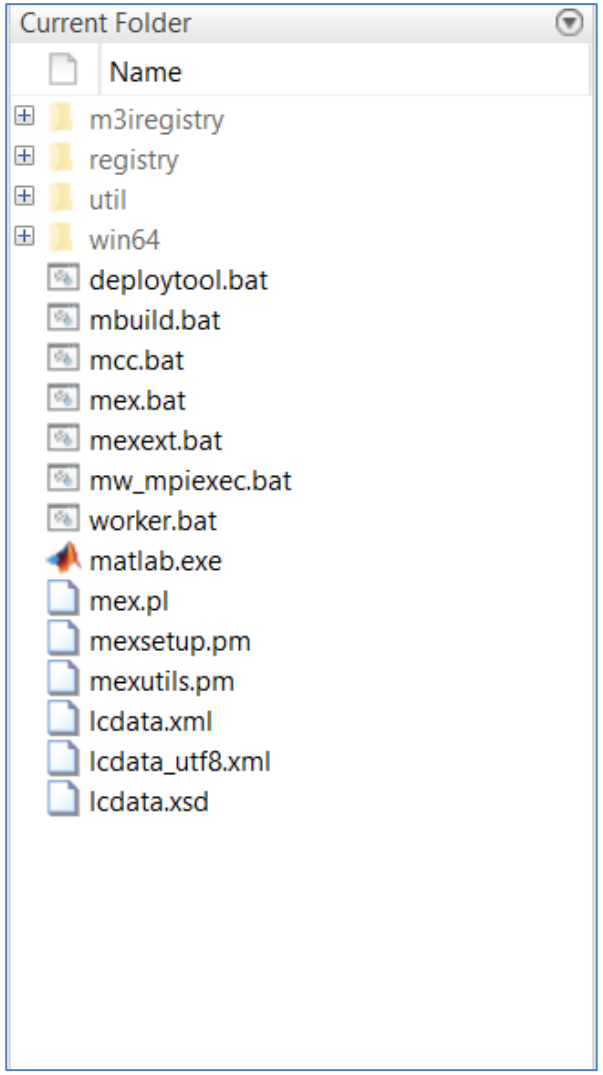
Advantages:

- Ease of Use,
- Operating system compatibility,
- Ease of numerical analysis,
- Ready functions,
- Ease of viewing (graphic drawing),
- Ease of GUI (Graphical User Interface) development,
- Toolboxes (toolboxes/ready programs).

Main features of MATLAB / USER INTERFACE

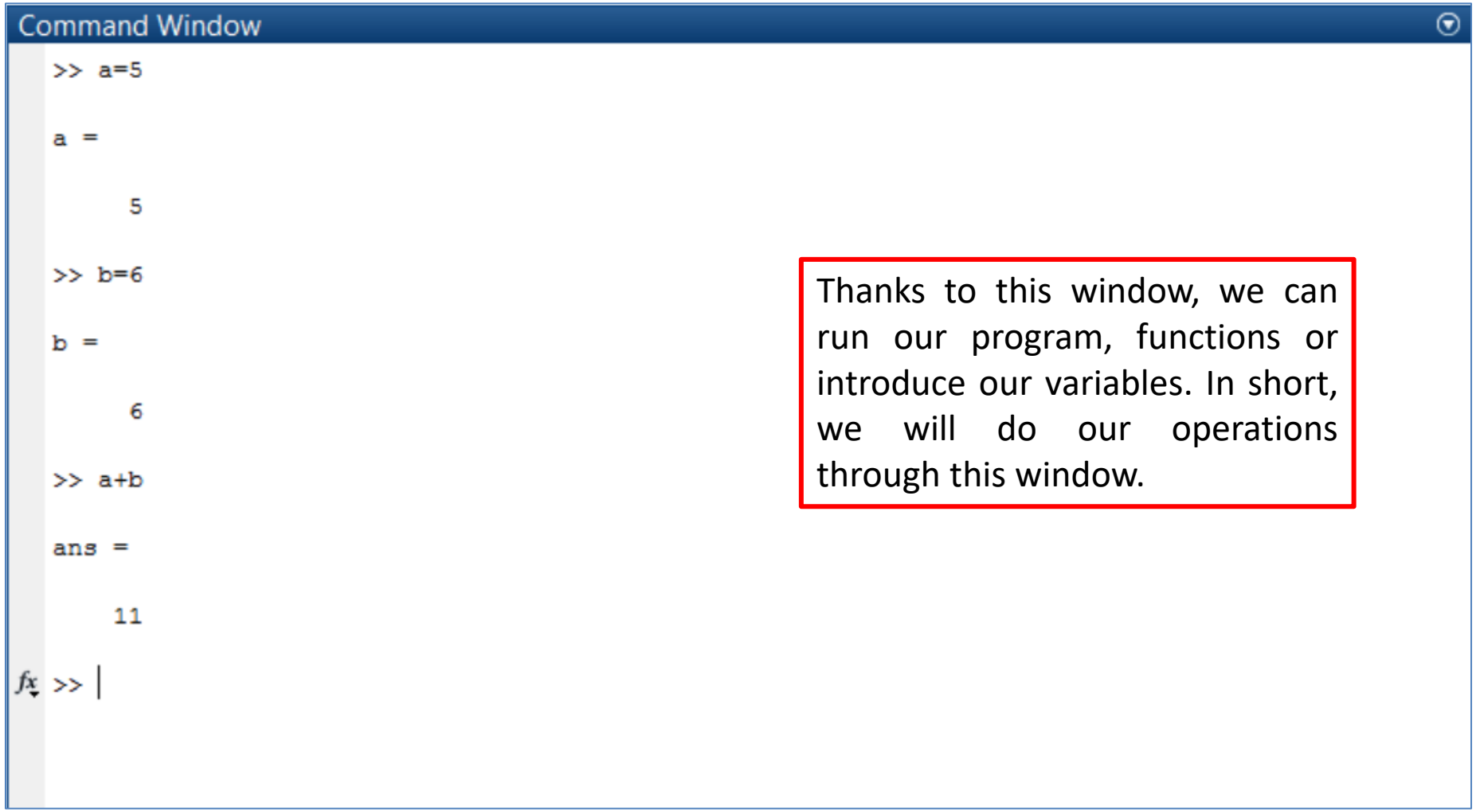


Main features of MATLAB / CURRENT FOLDER



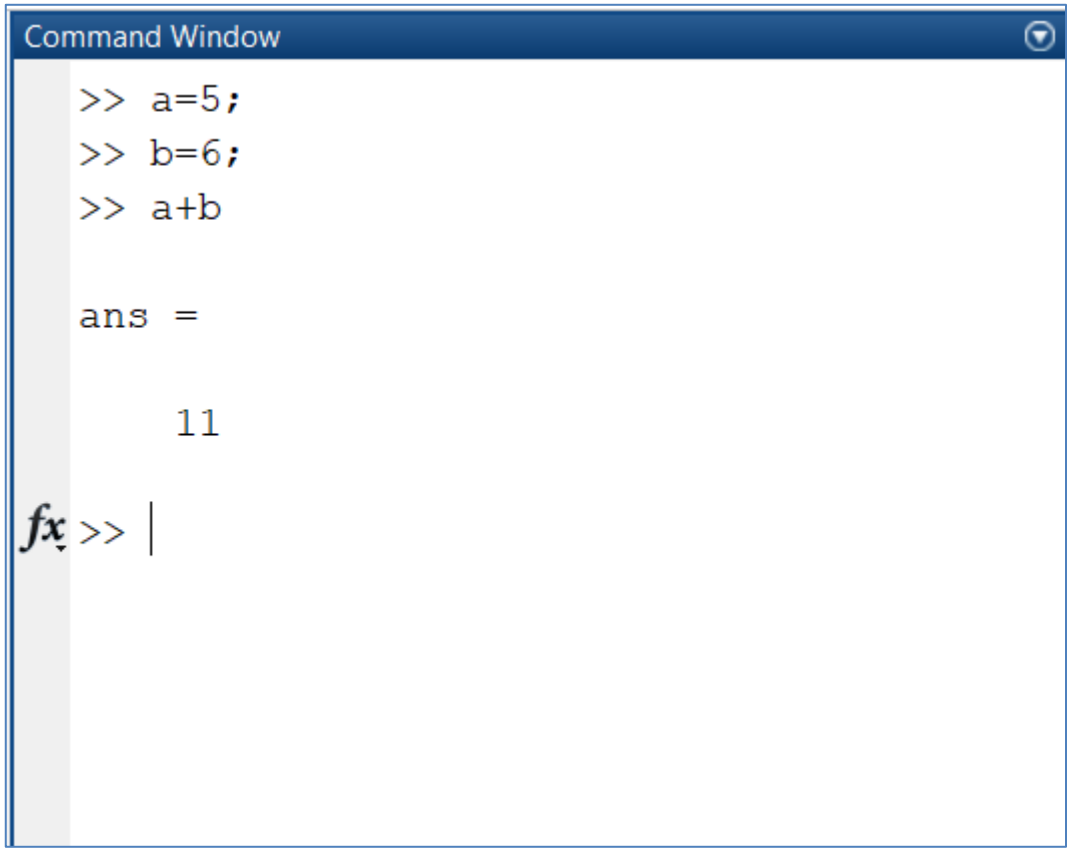
- This window shows our Work Folder.
- In other words, it is the folder where the files that will be required for the program we are working on are saved.
- We can change this folder whenever we want and easily access our data from here.

Main features of MATLAB / COMMAND WINDOW



Thanks to this window, we can run our program, functions or introduce our variables. In short, we will do our operations through this window.

Main features of MATLAB / COMMAND WINDOW



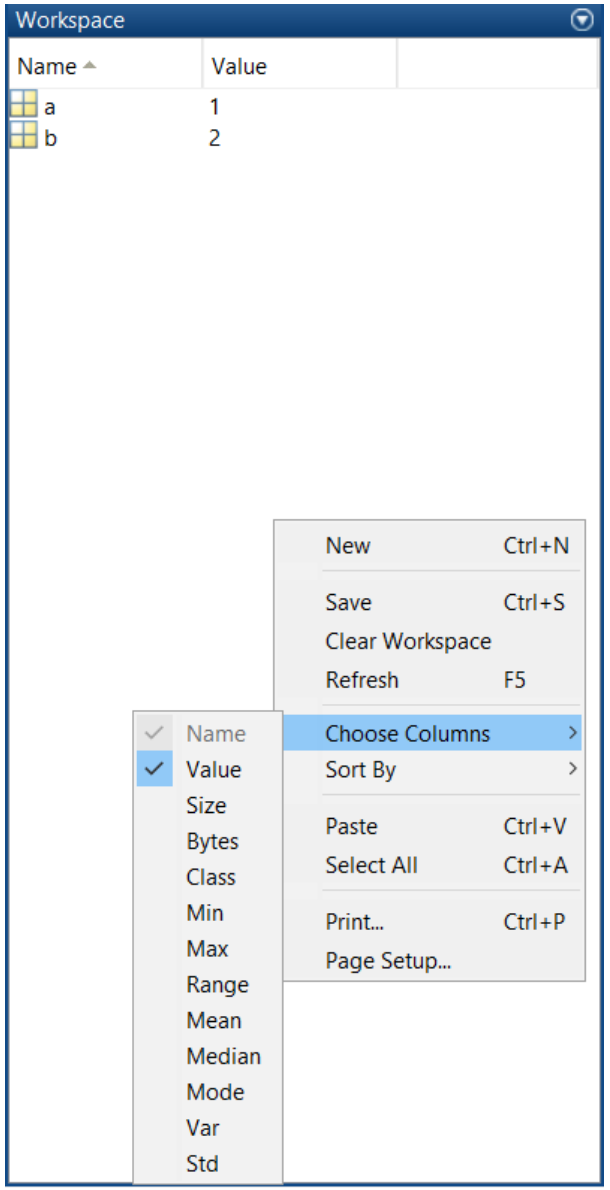
Let's do the same thing this time by putting a comma at the end of the variable and see the difference.
a=5; and b=6;
As you can see when we wrote it, it did not show the result on the screen.

Main features of MATLAB / COMMAND WINDOW

Correction of mistakes on the command line: The arrow keys on the keyboard allow to correct mistakes made on the command line. These are up "↑" down "↓" left "←" right "→". By using the up key, the previous line is displayed again, and editing is performed by moving the cursor to the wrong written place with the right and left keys.

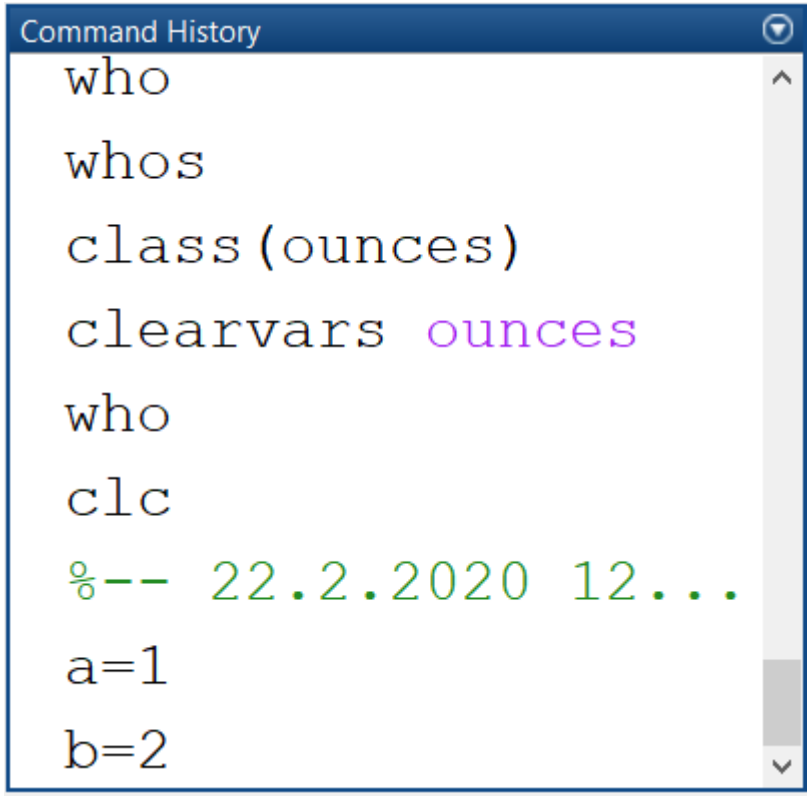
Hiding the Result from Displaying on the Screen: By typing an expression and pressing the Enter key, the results are automatically displayed on the screen. In contrast, If ";" is added, the calculations made with this expression will not be displayed on the screen.

Main features of MATLAB / WORKSPACE



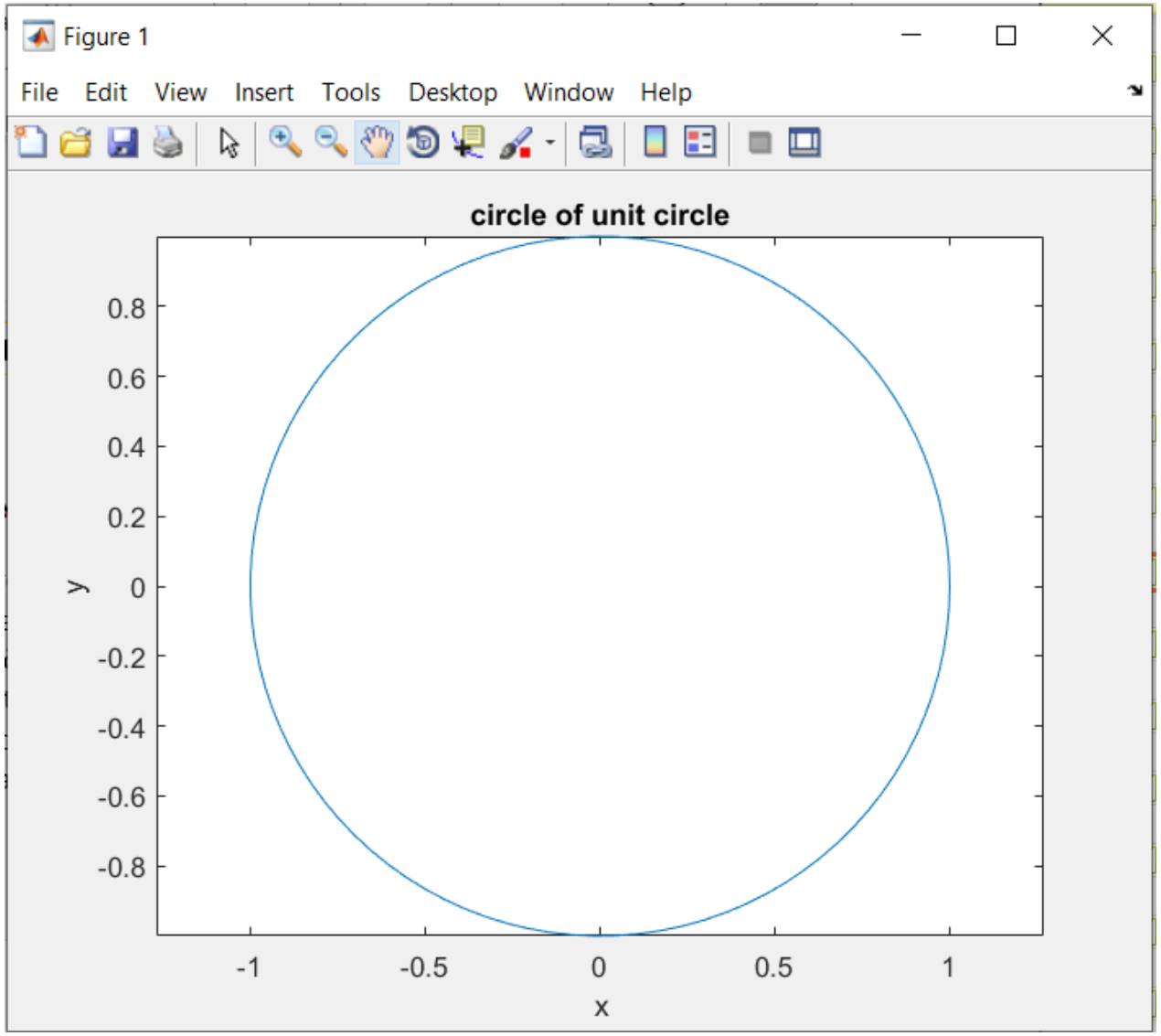
- This window displays information about the variables.
- We can see information such as name, value, size in this window.
- When we right-click on the window, we can select the information we want to see as seen in the picture below.

Main features of MATLAB / COMMAND HISTORY



- As the name suggests, this window shows the history of our commands.
- From this window, we can see the operations we have done retrospectively.

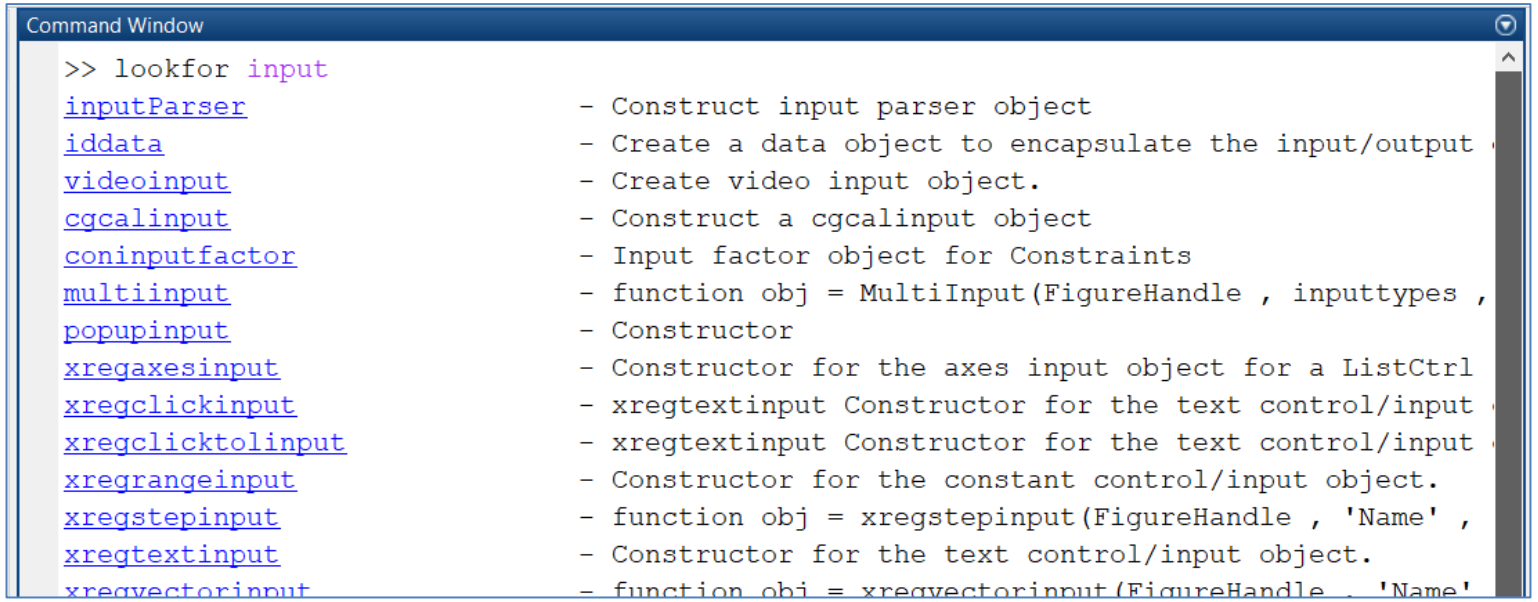
Main features of MATLAB / FIGURE WINDOW



Main features of MATLAB / GETTING HELP

- To find the appropriate function:

```
>> lookfor keyword
```



Main features of MATLAB / GETTING HELP

- For quick help with syntax and function definition:

```
>> help function
```

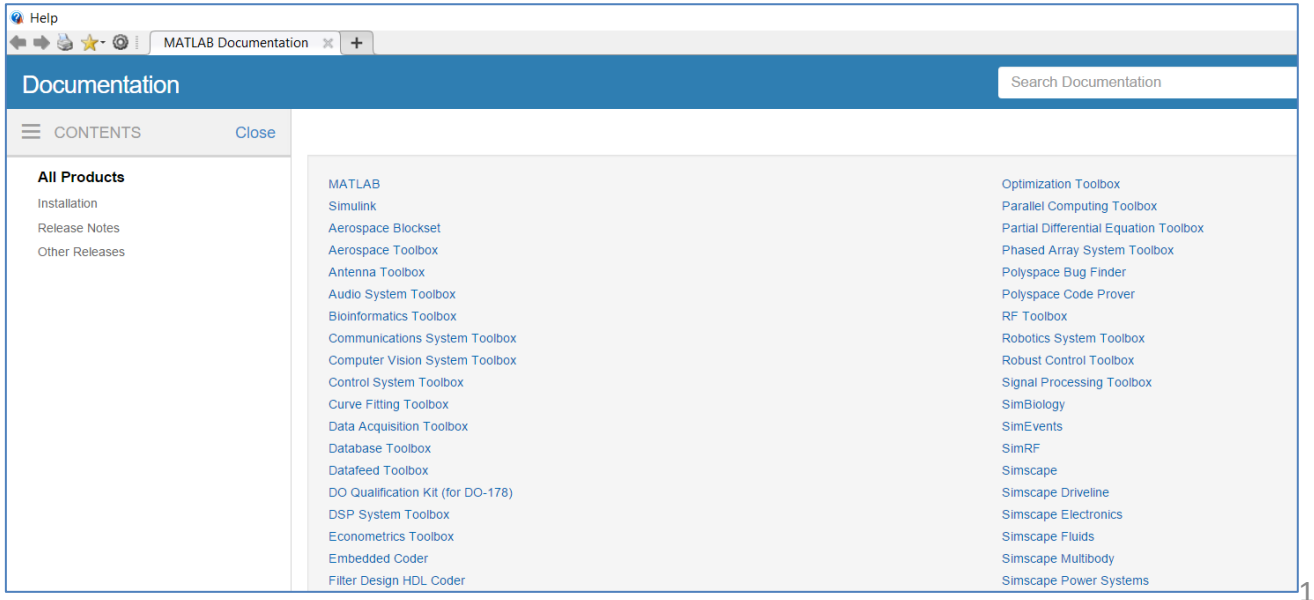
Command Window

```
>> help sin
 sin    Sine of argument in radians.
    sin(X) is the sine of the elements of X.

    See also asin, sind.

    Reference page for sin
    Other functions named sin
```

Main features of MATLAB / GETTING HELP

- For the advanced help system:

```
>>doc
```



Variables and Rules in MATLAB

As with many other programming languages, MATLAB requires mathematical expressions, but unlike many other programming languages, these expressions are completely represented by matrices.

Expression-forming groups: Variables, numbers, operators, and functions. MATLAB expressions are any command lines written in the command environment.

Variables: Expressions that replace numeric values within expressions.

- When MATLAB encounters a variable, it is automatically created, and enough memory is allocated.
- If the variable is already defined, MATLAB changes its contents and allocates new memory if necessary.

For example: >>x=50 %When this expression entered in the command line, the variable named x is created and 50 is assigned as a value of this variable.

- Like other computer languages, MATLAB has some rules about variable names!

Variables and Rules in MATLAB

Major Rules Can Be Summarized As Below:

1. Variable names are case sensitive. Accordingly, the words “high”, “High”, “hiGh” and “HIGH”, which mean the same but are spelled differently, are different variables for MATLAB.
2. Variable names can contain up to 31 characters. Anything more than that is ignored.
3. Variable names **must always start with a letter** and can be followed by any number of letters, numbers or underscore ”_”.
4. Punctuation marks are not used in variable names. Because many of them have a meaning in MATLAB.

Variables and Rules in MATLAB/ EXPRESSIONS

An expression is a formula consisting of variables, numbers, operators, and function names. It is evaluated when you enter it at the MATLAB prompt.

For example, evaluate 2π as follows:

```
2 * pi
```

MATLAB's response is:

```
ans =
    6.2832
```

Note that MATLAB uses the function ans (which stands for answer) to return the last expression to be evaluated but not assigned to a variable.

Variables and Rules in MATLAB/ STATEMENTS

MATLAB statements are frequently of the form

variable = expression

as in

```
s = u * t - g / 2 * t .^ 2;
```

This is an example of an *assignment* statement because the value of the expression on the right is *assigned* to the variable (s) on the *left*.
Assignment always works in this direction.
A common mistake is to get the statement the wrong way around, as in

```
a + b = c
```

Variables and Rules in MATLAB/ STATEMENTS

Basically, any line that you enter in the Command Window or in a program, which MATLAB accepts, is a statement, so a statement can be an assignment, a command, or simply an expression, such as

```
x = 29;          % assignment
clear            % command
pi/2             % expression
```

Variables and Rules in MATLAB

Numbers:

1. In MATLAB, numbers are expressed in the commonly used decimal base.
2. They can also be expressed in decimal exponential or in complex number formats as **i** or **j**.

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

3. In MATLAB, **i** and **j** represent the complex number **i**, unless otherwise defined.
4. The letter e (or E) in scientific notation represents the power of 10. The expression 3e5 means $3*10^5$.
5. In Matlab, all numbers range from approximately $2*10^{308}$ to $2*10^{-308}$.

Variables and Rules in MATLAB

The `format` command: The term `format` refers to how something is laid out: in this case MATLAB out-put. The default format in MATLAB has the following rules:

- It always attempts to display integers (whole numbers) exactly. However, if the integer is too large, it is displayed in scientific notation with five significant digits—1234567890 is displayed as `1.2346e+009` (i.e., 1.2346×10^{9})
- Numbers with decimal parts are displayed with four significant digits. If the value x is in the range $0.001 < x \leq 1000$, it is displayed in fixed-point form; otherwise, scientific (floating-point) notation is used. For example, 1000.1 is displayed as `1.0001e+003`

You can change from the default with variations on the `format` command, as follows.

Variables and Rules in MATLAB

COMMAND	FUNCTION
format short e	values displayed in scientific notation
format long e	values displayed in scientific notation but with 15 significant digits
format bank	values displayed in fixed point with two decimal digits (for cents)
format compact	gives a more compact display
format hex	gives hexadecimal display
format rat	display a number as a rational approximation

Variables and Rules in MATLAB

Table 2.1 Arithmetic operations between two scalars.

Operation	Algebraic form	MATLAB
Addition	$a + b$	`a + b`
Subtraction	$a - b$	`a - b`
Multiplication	$a \times b$	`a * b`
Right division	a/b	`a / b`
Left division	b/a	`a \ b`
Power	a^b	`a ^ b`

Variables and Rules in MATLAB

Table 2.2 Precedence of arithmetic operations.

Precedence	Operator
1	Parentheses (round brackets)
2	Power, left to right
3	Multiplication and division, left to right
4	Addition and subtraction, left to right

Arrays in MATLAB

The basic unit in MATLAB is arrays. An array is a structure that contains a certain number of values, structured using rows or columns. It is possible to collect the arrays in two groups:

- Vectors
- Matrices

Arrays in MATLAB / VECTORS

Vectors are arrays of just one column or just one row.

$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ a_n \end{bmatrix}$$

The element of a vector, after specifying the name of the directory, in parenthesis by typing the row and column numbers of the element (i.e. the element in the array by specifying its location) can be reached.

The expression `A(3)` corresponds to the 3rd element of the array `A`.

```
Command Window
>> A=[0;1;2;3;4]

A =

     0
     1
     2
     3
     4

>> A(3)

ans =

     2
```

Arrays in MATLAB / VECTORS

$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ a_n \end{bmatrix}$$

```
Command Window
>> A=[0;1;2;3;4]

A =

     0
     1
     2
     3
     4

>> A(3)

ans =

     2
```

A semicolon (;) indicates the end of the line.

Arrays in MATLAB / VECTORS

Command Window

```
>> A=[0 1 2 3 4]

A =

     0     1     2     3     4

>> A(2)

ans =

     1
```

Arrays in MATLAB / MATRICES

Matrices are arrays of multiple rows and columns.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nm} \end{bmatrix}$$

Command Window

```
>> A=[0 1 2 3; 4 5 6 7; 8 9 10 11]

A =

     0     1     2     3
     4     5     6     7
     8     9    10    11

>> A(2,4)

ans =

     7
```

The expression `A (2,4)` corresponds to the element in the 4th column of the 2nd row of the A array.

Arrays in MATLAB / MATRICES

```
Command Window
>> A=[0 1 2 3; 4 5 6 7; 8 9 10 11]

A =

     0     1     2     3
     4     5     6     7
     8     9    10    11

>> B=A(:,2)

B =

     1
     5
     9
```

The expression `A(:,2)` takes the 2nd column of matrix A and creates a new column matrix.

The colon operator: The colon operator has a lower precedence than the plus operator, as the following shows:

```
1+1:5
```

The addition is carried out first and a vector with elements 2, . . . , 5 is then initialized.

You may be surprised at the following:

```
1+[1:5]
```

The value 1 is added to *each element* of the vector 1:5. In this context, the addition is called an *array operation* because it operates on each element of the vector (array).

Variables and Rules in MATLAB

Table 2.3 Arithmetic operators that operate element-by-element on arrays.

Operator	Description
.*	Multiplication
./	Right division
.\	Left division
.^	Power

`linspace` **Function:** Creates an array given the initial value and the last value by linearly incrementing the given number of elements

```
Variable= linspace(initial value, final value, # of elements)
```

```
>> x=linspace(0,10,6)
x=
      0      2      4      6      8      10
```

`logspace` **Function:** The function `logspace` can be used to generate logarithmically spaced data.

```
>> y=logspace(0,2,10)
y=
      1.0000    1.6681    2.7826    4.6416    7.7426   12.9155
21.5443     35.9381   59.9484  100.0000
```

Variables and Rules in MATLAB / Assigning variable

Assigning Scalar

```
X   =   3
A   =   5-5i
B   =   A / 5
```

Assigning Vector

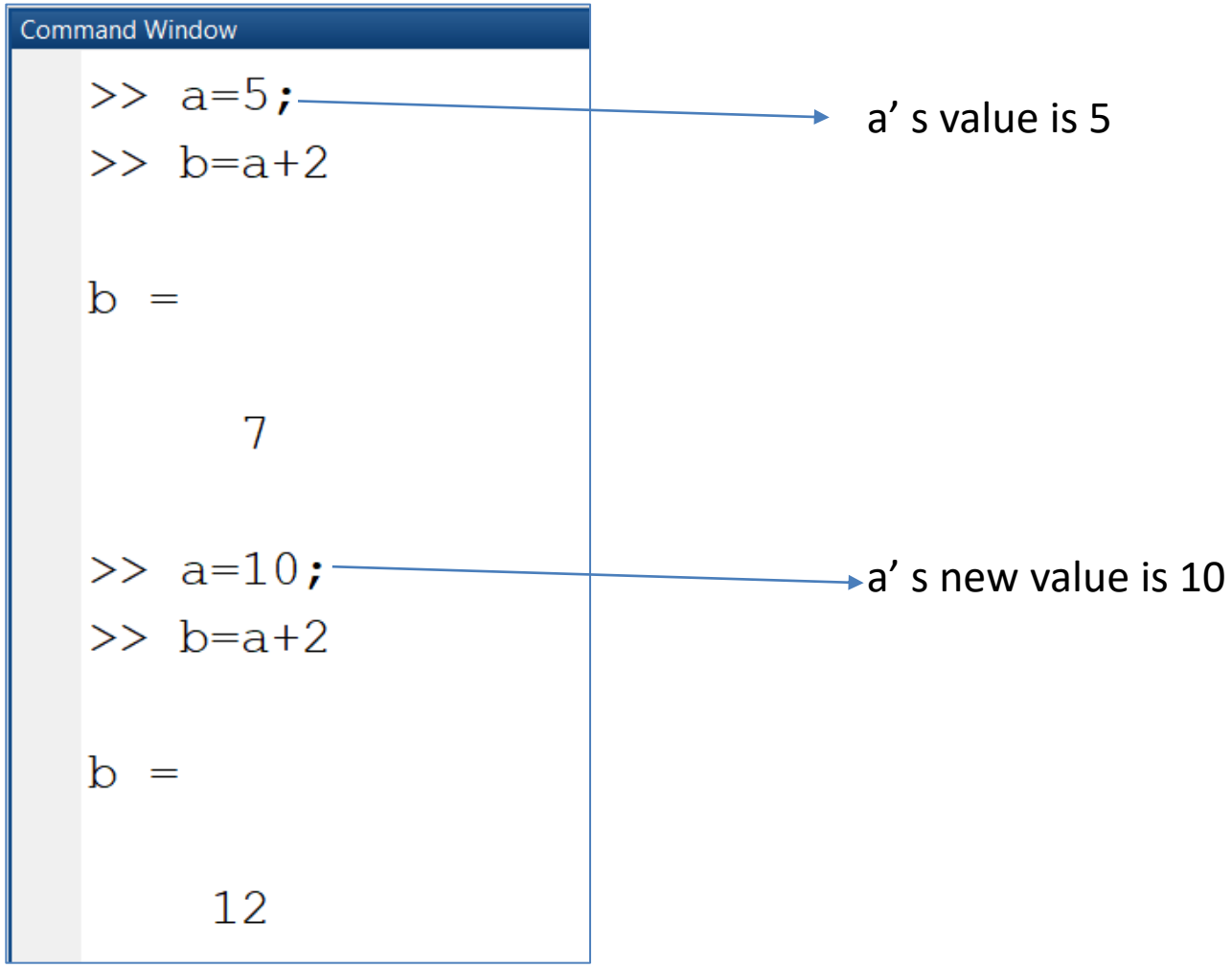
```
>>C=[1  3  2]
C =
     1      3
2

>> C= [1;3;2]
C =
     1
     3
     2
```

Assigning Matrix

```
>>C=[1  3; 2  1]
C =
     1      3
     2      1

>> C=[1, 3; 2,
1]
C =
     1      3
     2      1
```

After assigning a value to a variable, assigning a different value to the same variable causes the initial value of that variable to be deleted and the variable in question to be treated with its new value in subsequent operations.

Variables and Rules in MATLAB / Assigning variable



Variables and Rules in MATLAB / Assigning variable

Assigning Arrays Created Using MATLAB Ready Function Properties to Variables:

zeros(n)	Creates nxn zero matrix
zeros(n,m)	Creates nxm zero matrix
ones(n)	Creates nxn ones matrix
ones(n,m)	Creates nxm ones matrix
eye(n)	Creates nxn eye matrix
length(x)	Gives the column number of ‘ the x array’
size(x)	Gives the column and row number of ‘ the x array’

```
>> A=zeros(2)

  A =

     0     0
     0     0
```

```
>>B= ones(2,3)

  B =

     1     1     1
     1     1     1
```

```
>>C= eye(3,3)

  C =

     1     0     1
     0     1     0
     0     0     1
```

Variables and Rules in MATLAB / Assigning variable

Adding commonly used constants to the workspace: If you often use the same physical or mathematical constants in your MATLAB sessions, you can save them in an M-file and run the file at the start of a session. For example, the following statements can be saved in `myconst.m`:

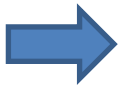
```
g = 9.81;                % acceleration due to gravity
avo = 6.023e23;          % Avogadro's number
e = exp(1);              % base of natural log
pi_4 = pi / 4;
log10e = log10( e );
bar_to_kP = 101.325;     % atmospheres to kiloPascals
```

If you run `myconst` at the start of a session, these six variables will be part of the workspace and will be available for the rest of the session or until you clear them.

Variables and Rules in MATLAB / Assigning variable

REQUESTING AN EXTERNAL ASSIGNMENT TO A VARIABLE: The `input` function displays a prompt in the command window asking the user to enter a value in a variable and waits for the user to enter that value,

```
>> x=input("Enter the x value: ")
Enter the x value:
```



```
>> x=input("Enter the x value: ")
Enter the x value: 5

x =

     5
```

Next week

Chapter 4

VECTOR AND MATRIX OPERATIONS