**Chapter 1 Starting with MATLAB**

**Introduction**

MATLAB is a powerful language for technical computing. The name MATLAB stands for MATrix LABoratory, because its basic data element is a matrix (array). MATLAB can be used for math computations, modeling and simulations, data analysis and processing, visualization and graphics, and algorithm development. MATLAB is widely used in universities and colleges in introductory and advanced courses in mathematics, science, and especially engineering. In industry the software is used in research, development, and design. The standard MATLAB program has tools (functions) that can be used to solve common problems. In addition, MATLAB has optional toolboxes that are collections of specialized programs designed to solve specific types of problems. Examples include toolboxes for signal processing, symbolic calculations, and control system.

Until recently, most of the users of MATLAB have been people with previous knowledge of programming languages such as FORTRAN and C who switched to MATLAB as the software became popular. Consequently, the majority of the literature that has been written about MATLAB assumes that the reader has knowledge of computer programming. Books about MATLAB often address advanced topics or applications that are specialized to a particular field. Today, however, MATLAB is being introduced to college students as the first (and often the only) computer program they will learn. For these students there is a need for a book that teaches MATLAB assuming no prior experience in computer programming.

The MATLAB environment consists of the following five major ingredients:

(1) MATLAB Language: This is a high‐level matrix (array) language with control flow programming features. It allows the user to program in the small (creating throw‐away programs) and program in the large (creating complete large and complex application‐specific statements, functions, data structures, input/output, and object‐oriented programs).

(2) MATLAB Working Environment: This is a set of tools and facilities. It includes facilities for managing the variables in workspace, manipulation of variables and data, importing and exporting data, etc. Tools for developing, managing, debugging, and profiling m‐files for different applications are available.

(3) Handle Graphics@: This is the MATLAB graphics system. It includes high‐level commands for two‐ and three‐dimensional data visualization, image processing, animation, and presentation. It also includes low‐level commands that allow the user to fully customize the appearance of graphics and build complete graphical user interfaces (GUIs).

(4) MATLAB Mathematical Function Library: This is a collection of computationally efficient and robust algorithms and functions ranging from elementary functions (sine, cosine, tangent, cotangent, etc.) to specialized functions (Eigen values, Bessel functions, Fourier and Laplace transforms, etc.) commonly used in scientific and engineering practice.

(5) MATLAB Application Program Interface (API): This is a library that allows the user to write C and FORTRAN programs that interact within the MATLAB environment. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB for computing and processing, reading and writing m‐files, etc. Real‐Time Workshop@ allows the user to generate C code from block diagrams and to run it for real‐time systems.

**1.1 Starting MATLAB, MATLAB Windows**

It is assumed that the software is installed on the computer, and the user can start the program in two ways:

1) From Start menu, then All Programs choose MATLAB 6.5 program.

2) Double‐click on the MATLAB icon  on the desktop.

يمكن بدء برنامج Matlab بأحدى الطريقتيين التاليتين: من قائمة start نذهب الى All programs ونختار برنامج Matlab أو بالنقر المزدوج على ايقونة Matlab الموجودة على سطح المكتب.

When you start MATLAB, you will briefly see a window that displays the MATLAB logo as well as some MATLAB product information, and then a MATLAB Desktop window will launch (figure (1.1)). That window will contain a title bar, a menu bar, a tool bar, and four smaller windows which are the Command Window on the right, the Workspace Browser, the Current Directory Window (top left), and the Command History Window (lower left). A list of the several windows and their purpose is given in table (1.1). At the top of the Command Window, you may see some general information about MATLAB, perhaps some special instructions for getting started and a line that contains a prompt. The prompt will likely be a double caret (>> ) which means the program interpreter is waiting for you to enter instructions. (Remember to press the Return key at the end of each line that you enter.). The line with the >> prompt is called the command line. If a window is “active”, its title bar will be dark.

Each of the windows can be closed by clicking on  the button. Any hidden or closed window can be launched through the View menu button (figure (1.2)).

You can separate windows inside your MATLAB Desktop out to your computer desktop by clicking on the curved arrow  in the upper right‐hand corner (you can return it to the Desktop by selecting **Dock** from the **View** menu of the undocked window). The **Start** button lower left side can be used to access MATLAB tools and features (figure (1.3)).

**Title Bar**

**شريط العنوان**

**Menu Bar**

**شريط القوائم**

**Tool Bar**

**شريط الادوات**

**Current Directory**

**الدليل الحالي**

**Command Window**

**نافذة الايعازات**



**Workspace Window**

**نافذة ساحة العمل**

Current Directory Window

نافذة الدليل الحالي

**Command History Window**

**نافذة تاريخ العمل**

**State Bar**

**شريط الحالة**

**Figure (1.1): The default view of MATLAB desktop**

****

**اختيار نافذة**

**نموذج سطح المكتب**

**Figure (1.2): The View menu**

****

**Figure (1.3): The Start button contents**

يمكن فصل النوافذ الجزئية في سطح مكتب **Matlab** إلى سطح المكتب للحاسوب بالضغط على السهم المعقوف اعلى اليمين  (و يمكن ارجاعها باختيار **Dock** من **View** ). يمكن استخدام زر **Start** أسفل اليسار للوصول الى ادوات و خواص ماتلاب.

|  |  |
| --- | --- |
| Purpose | Window |
| Logs commands entered in the Command Window. | Command History Window |
| Main window, enters variables, runs programs. | Command Window |
| Shows the files in the current directory. | Current Directory Window |
| Creates and debugs scripts and function files. | Editor Window |
| Contains output from graphic commands. | Figure Window |
| Provides help information. | Help Window |
| Provides access to tools, demos, and documentation. | Launch Pad Window |
| Provides information about the variables that are used. | Workspace Window |

**Table (1.1): MATLAB Windows**

**1.2 Entering Commands**

Every command has to be followed by a carriage return <cr> (**Enter** key) in order that the command can be executed. MATLAB commands are case sensitive and lower case letters are used throughout.

يجب الضغط على مفتاح Enter بعد كل إيعاز لتنفيذه. الايعازات حساسة لحالة الحروف وتكتب بحروف صغيرة.

**1.3 Aborting MATLAB**

If MATLAB is hung up in a calculation, or is just taking too long to perform an operation,you can usually abort it by using the keys **Ctrl+C.**

احيانا يعلق Matlab في الحسابات أو يستغرق وقتاً طويلاً في العمليات فلقطع تنفيذ البرنامج نضغط **Ctrl+C.**

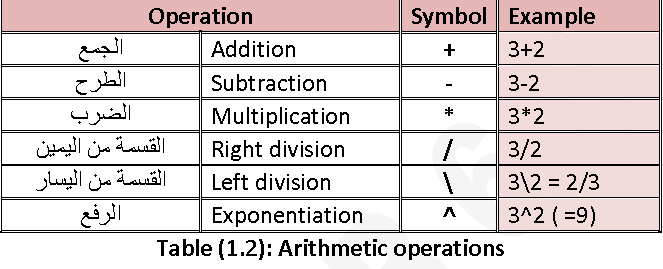
**1.4 Quitting MATLAB**

In order to quit MATLAB, type **quit** or **exit** after the prompt. You can also click on the special symbol that closes your windows (usually  in the upper left‐hand corner). Either of these may or may not close all the other MATLAB windows that are open. You may have to close them separately. Indeed, it is our experience that leaving MATLAB‐generated windows around after closing the MATLAB Desktop may be hazardous to your operating system. Still another way to exit is to use the **Exit MATLAB** option from the **File menu** of the Desktop. Before you exit MATLAB, you should be sure to save any variables, print any graphics or other files you need, and in general clean up after yourself.

الخروج النظامي من Matlab عبر اختيار Exit Matlab من قائمة File او بكتابة Exit or quit بعد الحاث prompt .

**1.5 Arithmetic Operations**

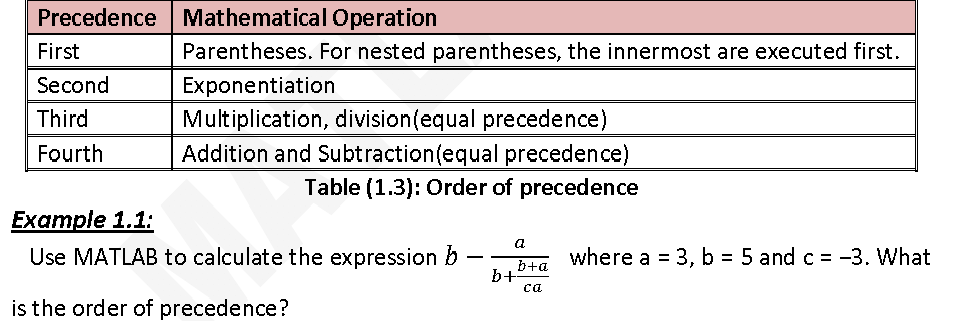
The symbols of arithmetic operations are:



**1.6 Order of Precedence**

MATLAB executes the calculations according to the order of precedence displayed

below. This order is the same as used in most calculators.



**Solution:**

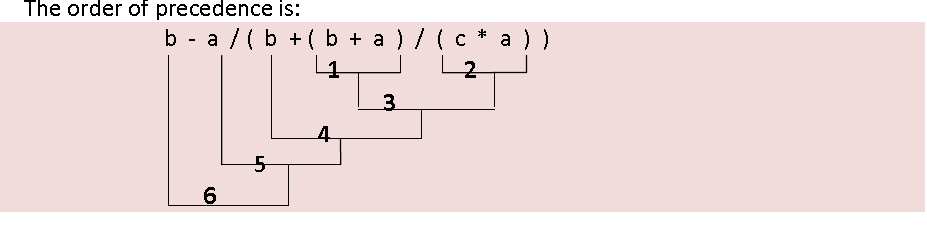
The code for this purpose is:

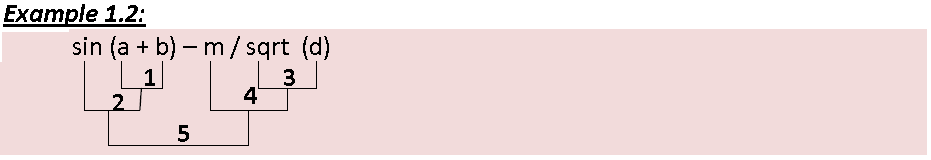
>> a = 3; b = 5; c = ‐3;

>> x = b‐a/ (b+ (b+a)/(c\*a))

x =

4.2703





**1.7 Format**

MATLAB uses double‐precision floating point arithmetic, which is accurate to approximately 15 digits; however, MATLAB displays only 5 digits by default (short loss). The user can control the format in which MATLAB displays output on the screen. The format can be changed with the **format** command. Once the **format** command is entered, all the output that follows is displayed in the specified format. The **format** function affects only how numbers are displayed, not how MATLAB computes and saves them. There are several of available formats (see laboratory guide).

يستخدم ايعاز Format للتحكم بشكل المخرجات على الشاشة وليس بالطريقة التي يحسبها بها Matlab او يخزنها.

**1.8 Defining Variables**

A variable is a name made of a letter or a combination of letters (and digits) that is assigned a numerical value. Once a variable is assigned a numerical value, it can be used in mathematical expressions, in functions, and in any MATLAB statements and commands. A variable is actually a name of a memory location. When a new variable is defined, MATLAB allocates an appropriate memory space where the variable is stored. When the variable is used the stored data is used. If the variable is assigned a new value the content of the memory location is replaced.

المتغير هو اسم مؤلف من حروف وأرقام يسند له قيمة عددية وعندما يعرف المتغير يخزنه Matlab في حيز ملائم في الذاآرة وحينها يمكن استخدامه في تعبيرات رياضية, في الدوال وفي أي تعبير أو إيعاز.

**1.8.1 Rules About Variable Names**

A variable can be named according to the following rules:

**•** Must begin with a letter. For example: 5xr, a\*b, \_gt, a‐b are not allowed as variable names.

• Can be up to 63 characters.

• Can contain letters, digits and the underscore ( \_ ).

• Cannot contain punctuation characters (e.g. period, comma, and semicolon).

• MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. That is: Ab, AB, aB, and ab are names for different variables.

• No spaces are allowed between characters.

• Avoid using the name of a built‐in function for a variable name. Once a function name is used to define a variable, the function cannot be used.

• It is good programming practice to employ meaningful variable names and the variable names to be informative.

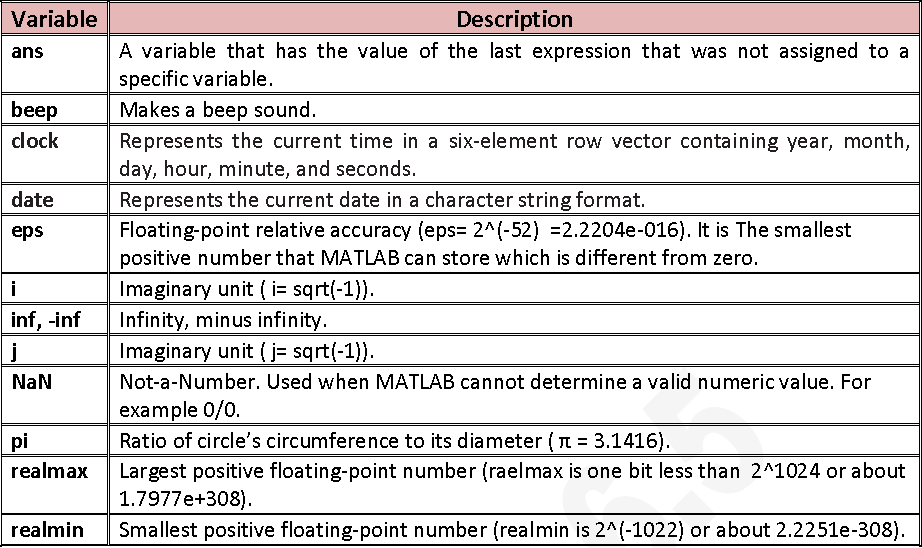
**1.8.2 Predefined Variables and Keywords**

There are several words, called keywords, which are reserved by MATLAB for various purposes, and cannot be used as variables names. These words are:

**Break case catch continue else elseif end for function global if Otherwise persistent return switch try while**

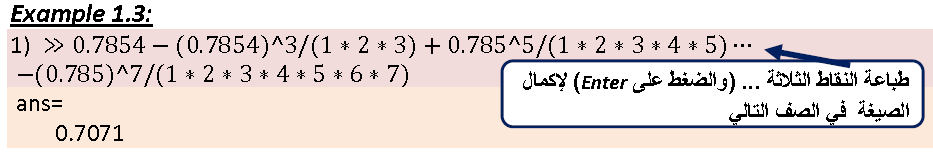
When typed, these words appear in blue. An error message is displayed if the user tries to use a keyword as a variable name (the keywords can be displayed by typing the command **iskeyword**).

A number of frequently used variables are already defined when MATLAB is started. The predefined variables are:



**1.9 Notes for Working in the Command Window**

* Once a command is typed and the Enter key is pressed, the command is executed. However, only the last command is executed. Everything executed previously is unchanged.
* Several commands can be typed in the same line separated by commas ( , )or semicolons ( ; ) . The commands are executed in order from left to right.
* It is not possible to go back to a previous line in the Command Window, make a correction, and then re‐execute the command.
* A previously typed command can be recalled to the command prompt with the uparrow key ( ↑ ). When the command is displayed at the command prompt, it can be modified if needed and executed. The down‐arrow key ( ↓ ) can be used to move down the previously typed commands.
* If a command is too long to fit one line, it can be continued to the next line by typing three periods ( … )(called an ellipsis) and pressing the Enter key. The continuation of the command is the typed in the new line. The command can continue line after line up to a total of 4096 characters.
* MATLAB has a useful editing feature called **smart recall**. Just type the first few characters of the command you want to recall, e.g. type the characters 2\* and press the Up‐arrow key—this recalls the most recent command starting with 2\*.



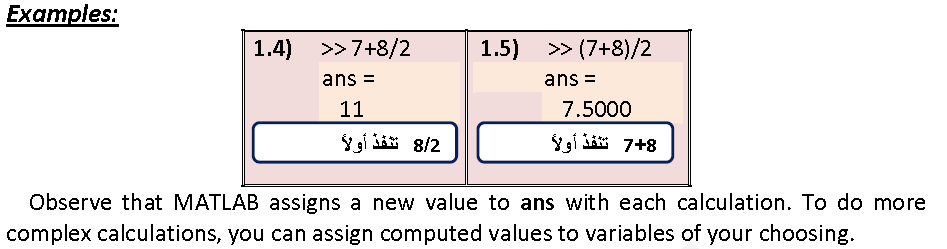
**1.10 The Semicolon ( ; )**

When a command is executed, any output that the command generates is displayed in the Command Window. If the semicolon ( ; ) is typed at the end of a command the output of the command is not displayed. Typing a semicolon is useful when the result is obvious or known, or when the output is very large.

عندما تطبع الفارزة المنقوطة ; في نهاية أيعاز فان نتيجة ذلك الايعاز لا تظهر على الشاشة.

**1.11 Using MATLAB as a Calculator**

The simplest way to use MATLAB is as a calculator. This is done in the Command Window by typing a mathematical expression and pressing the Enter key. MATLAB calculates the expression and assigns the value to a variable called ans. If you want to perform further calculations with the answer, you can use the variable ans rather than retype the answer.

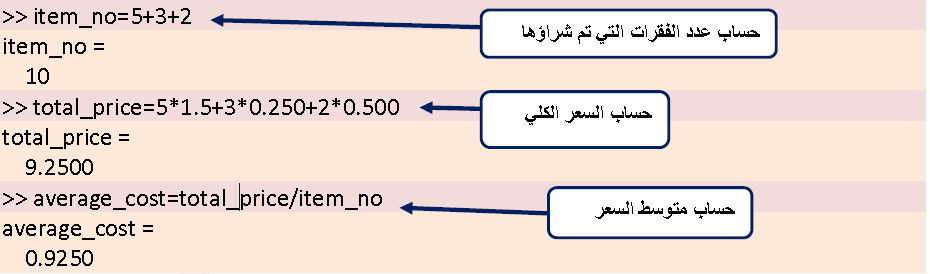


تستخدم ans كاسم متغير

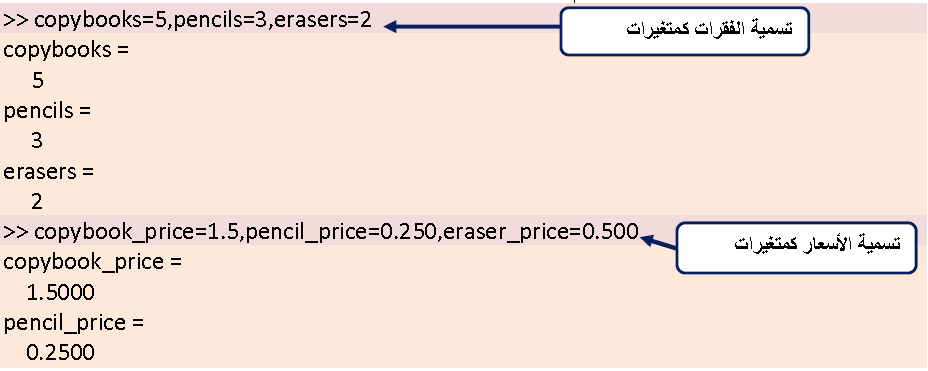
**Example 1.6:**

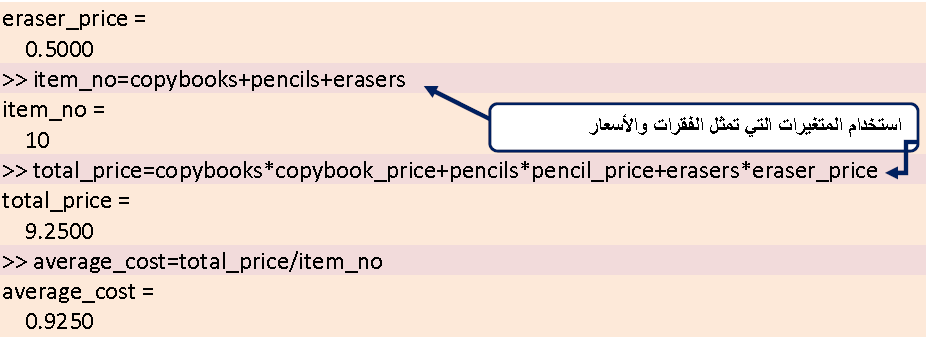
If a person buys 5 copybooks, 3 pencils and 2 erasers. Their prices are: 1.5, 0.250 and 0.500 ID for each unit, respectively. How many item does this person buys, what is the total price and average cost?

**Solution:**

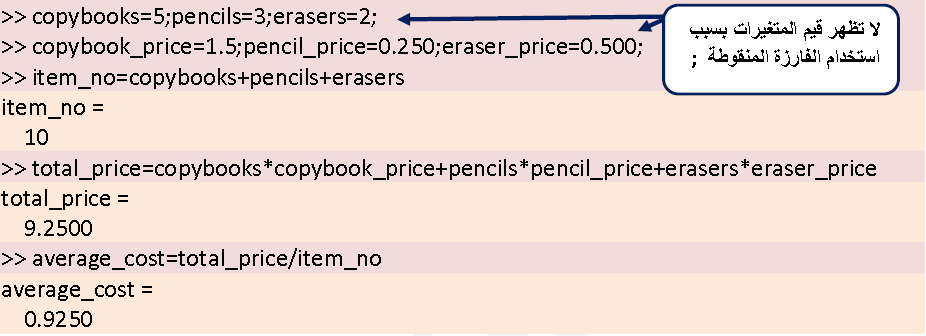
****

We can solve this example in another way by giving a name to each variable , also we can use more than one command in each line as follows:





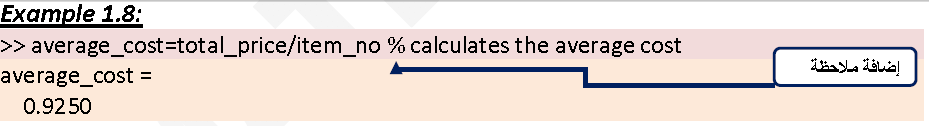
Also, we can hide input and show only output using semicolons:

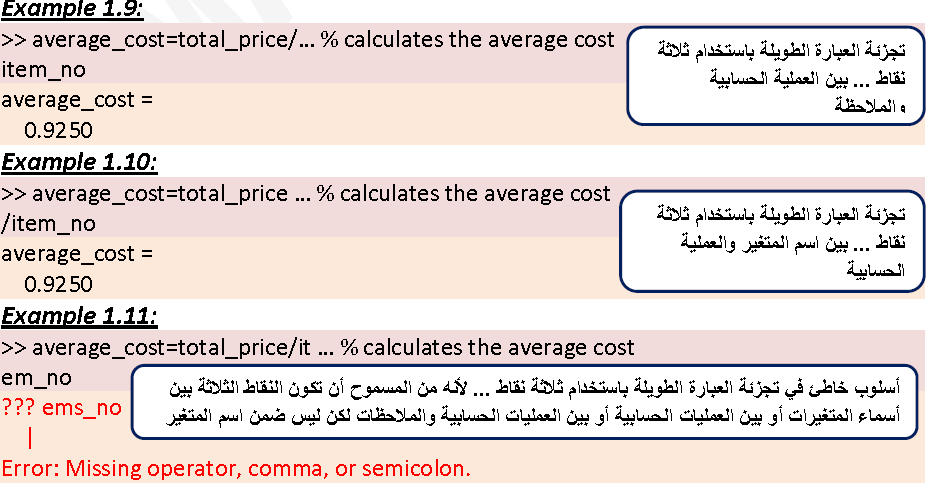


**1.12 Typing %**

When the symbol ( % ) (percent symbol) is typed in the beginning of a line, the line is designed as a comment, that is the line is not executed. The % character followed by a text can also be typed after a command; it has no effect on the execution n of the command.

الرمز % يستخدم لإضافة الملاحظات والعبارة التي تكتب بعده يعتبر ملاحظة وليس عبارة تنفيذية.





**1.13 who,whos**

The command who shows a list of all defined variables. The command whos shows information about all defined variables (name, size, type), but it does not show the values of the variables (see laboratory guide).

**1.14 The clc Command**

The clc command clears the Command Window (see laboratory guide).

**1.15 The clear Command**

The clear command clears the memory (see laboratory guide). You should generally clear variables before starting a new calculation. Otherwise values from a previous calculation can creep into the new calculation by accident.

**1. 16 The clf Command**

This command creates a graph window (if one does not already exist) or clears an

existing graph window.

الايعاز clear يستخدم لتنظيف الذاكرة (وليس الشاشة) بينما الإيعازclc يستخدم لتنظيف الشاشة (وليس الذاكرة).اما الايعاز clf فيستخدم لتكوين شاشة رسوميات (إذا لم تكن موجودة) أو لحذف شاشة الرسوميات الموجودة.

**1.17 The Keys Tab and Esc**

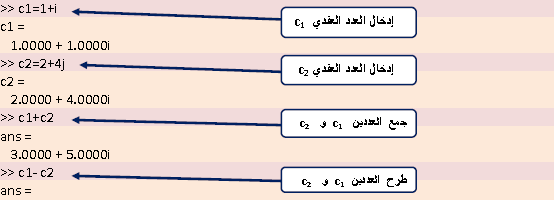
The Tab key is used to auto finish functions name. The **Esc** key (Escape) delete the whole expression written after the prompt.

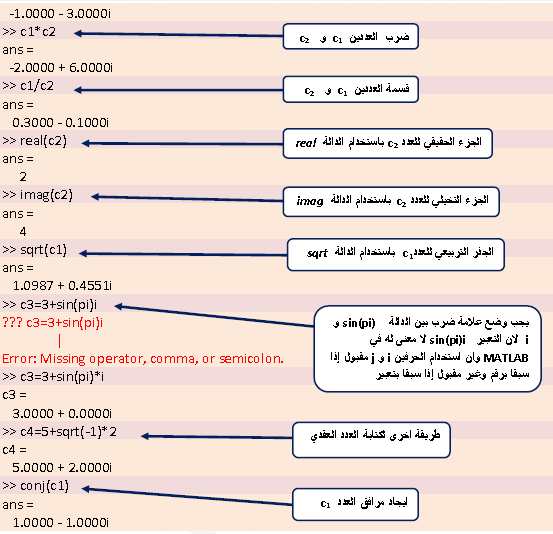
**1.18 Complex numbers**

MATLAB does not need any special procedures when managing complex numbers and apply arithmetic operations on them similarly to real numbers. The letters i and j are used to denote (So as not to confuse the usual symbol for a current with this quantity, electrical engineers prefer the use of the j symbol. MATLAB accepts either symbol, but always gives the answer with the symbol i). Real part, imaginary part, conjugate, and magnitude (modulus) of a complex number can be found using the functions: **real(x)**, **imag(x)**, **conj(x)**, and **abs(x)** respectively.

We all remember Euler’s rule to change the complex number from polar form to Cartesian form. The complex number z can be written in the following ways: z=(a, b)=a+ib = (r,θ) = r cosθ +i sinθ, where r= (called the **magnitude** of z) and θ = (called the **argument** of z). The angle of z can be found by the function **angle(z)**.

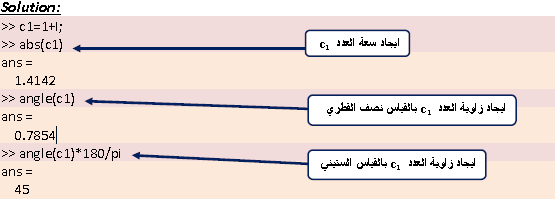
**Example 1.14:**

****



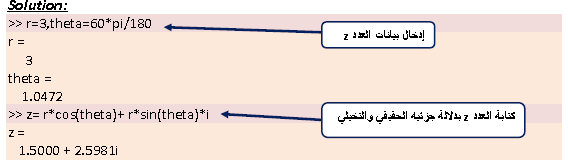
**Example 1.15:**

Find the magnitude and argument (in degrees) of c1=1+i.



**Example 1.16:**

Write the complex number z=(3,60°) in the Cartesian form.



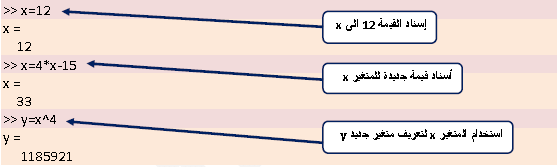
**1.19 The Assignment Operator**

In MATLAB the = sign is called the assignment operator. The assignment operator assigns a value to a variable

variable\_name=A numerical value, or a computable expression

When the **Enter** key is pressed the numerical value of the right‐hand side is assigned to the variable, and MATLAB displays the variable and its assigned value in the next two lines. If a variable already exists, it can be reassigned a new value. Once a variable is defined it can be used as an argument in functions.

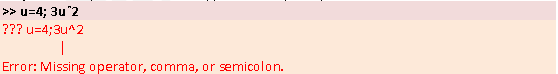
**Example 1.17:**



If a variable already exists, typing the variable name and pressing the Enter key willdisplay the variable and its value in the next two lines.

**1.20 Errors in input**

If you make an error in an input line, MATLAB will beep and print an error message (in red). For example, here’s what happens when you try to evaluate **3uˆ2**:



The error is a missing multiplication operator \*. The correct input would be **u=4**; **3\*uˆ2**. Note that MATLAB places a marker (a vertical line segment) at the place where it thinks the error might be; however, the actual error may have occurred earlier or later in the expression. Missing multiplication operators and parentheses are among the most common errors. Every variable is a matrix even scalar, they are considered 1 x 1 matrices.

**1.21 Online Help**

There are several ways to get online help in MATLAB. To simplest way to open the **Help Window** is by open it from **Help** menu, using any of the commands **doc** or **helpdesk**, or simply by clicking on  on the toolbar. We can also ask for help on certain function or toolbox (see laboratory guide).