Al-Maarif University College
Computer Engineering Techniqes
Second Class

Computer Applications

Prepared by

Aymen Jalil Abdulelah

2023 - 2024

Lecture One

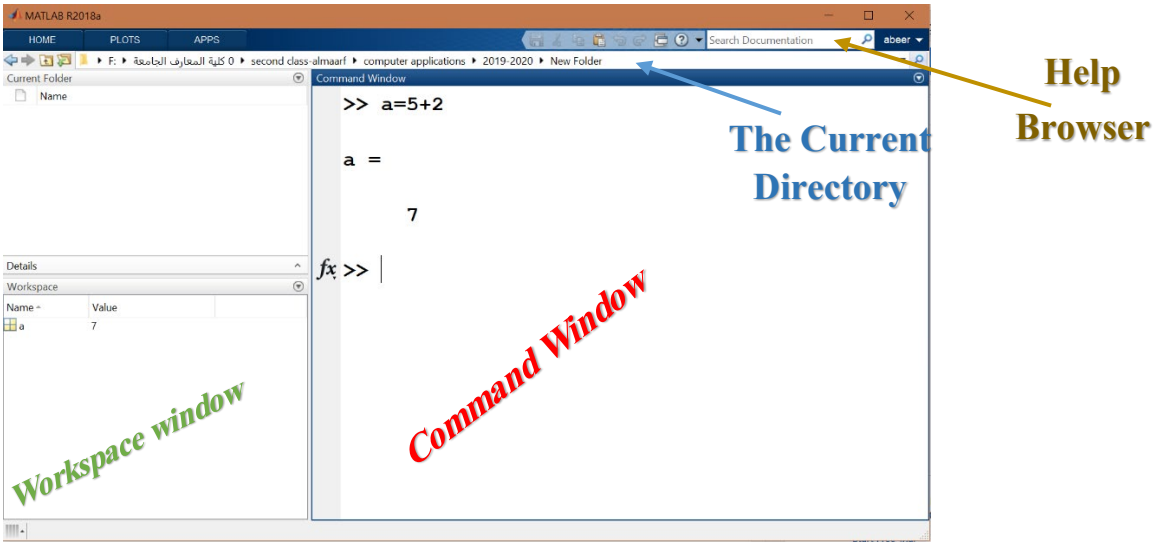
Basics of MATLAB

Introduction

MATLAB, short for MATrix LABoratory is a programming package specifically designed for quick and easy scientific calculations. It has literally hundreds of built-in functions for a wide variety of computations and many toolboxes designed for specific research disciplines, including statistics, optimization, solution of partial differential equations, data analysis. You need a solid knowledge of basic MATLAB commands and several more advanced features including two- and three-dimensional graphics, solution of algebraic equations, solution of ordinary differential equations, calculations with matrices and solutions of linear systems of equations.

Starting MATLAB

Start MATLAB by double-clicking on the MATLAB icon in the applications folder, or wherever it is. The MATLAB window should come up on your screen. It looks like:



This is the window in which you interact with MATLAB. The main window on the right is called the **command window**. You can see the command prompt in this window, which looks like >>. If this prompt is visible MATLAB is ready

for you to enter a command. In the figure, you can see that we typed in the command **a=5+2**.

In the lower left corner you can view the **workspace window**. The *Workspace window* will show you all variables that you are using in your current MATLAB session. In this example, the workspace contains the variable 'a'. When you first start up MATLAB, the workspace is empty.

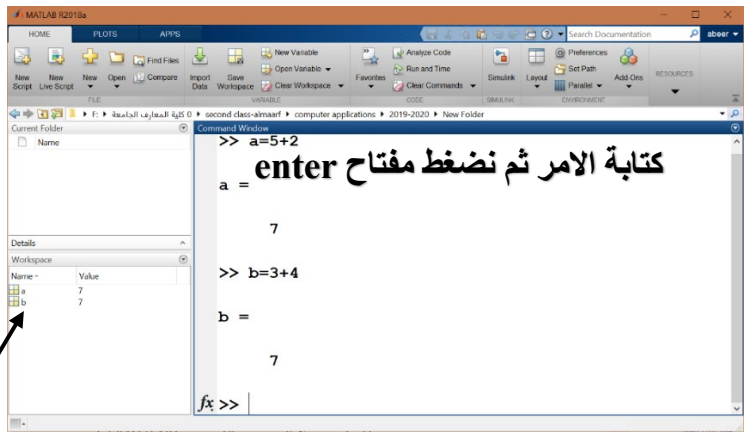
To change the layout of the MATLAB window, select Home, then Layout. Four different layout styles can be chosen.

Now, let's try a simple command. Next to the prompt in the ***Command Window type b=3+4*** and then press ***'enter'*** to activate this command. On the screen you should see

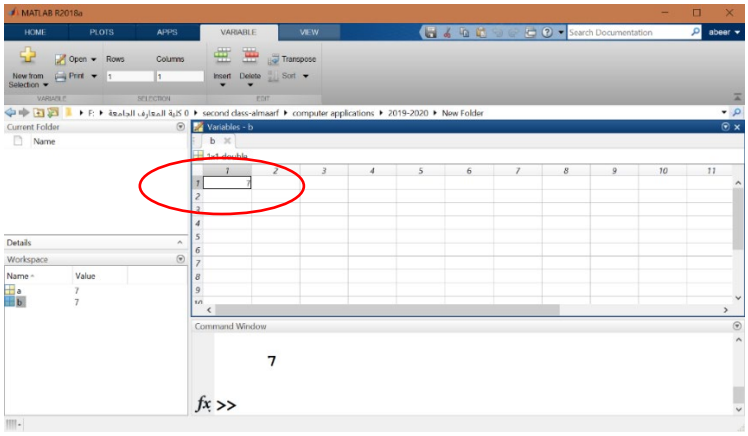
b=

7

Notice how MATLAB carries out this command immediately, and gives you the prompt >> for your next command. Here, a variable called 'b' is created by MATLAB and assigned the value of 7. MATLAB stores the variable a in its workspace until you exit MATLAB or tell MATLAB to delete the variable.



Go to the Workspace window and check that the variable b is in your workspace. You will see in this window that b is stored in 8 bytes, that it is a double and that it has size 1x1.



Useful Commands

- **clc**: to clean the command window
- **Clear**: to clean the workspace
- **Close**: to close any opened figure

Using MATLAB as a calculator

let's suppose you want to calculate the expression, 4 + 2 * 3. You type it at the prompt command (>>) as follows,

```
>> 4+2*3
ans =
    10
```

You will have noticed that if you do not specify an output variable, MATLAB uses a default variable **ans**, short for answer, to store the results of the current calculation. Note that the variable **ans** is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name. For example,

```
>> x =4 + 2 * 3
x =
10
```

will result in **x** being given the value 4 + 2 * 3 = 10. This variable name can always be used to refer to the results of the previous computations. Therefore, computing 4*x* will result in.

```
>> 4*x
ans =
    40
```

Table 1: Basic arithmetic operations

SYMBOL	OPERATION	EXAMPLE
+	Addition	$2+3$
−	Subtraction	$2-3$
*	Multiplication	$2*3$
/	Division	$2/3$

Creating MATLAB variables

MATLAB variables are created with an assignment statement. The syntax of variable as-assignment is

variable name = a value (or an expression)

For example,

```
>> x = expression
```

where expression is a combination of numerical values, mathematical operators, variables, and function calls. On other words, expression can involve:

- manual entry
- built-in functions
- user-defined functions

Overwriting variable

Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a semicolon (;) at the end of the line. Then the sequence of commands looks like this:

```
>> t = 5;
>> t = t+1
t =
6
```

Error messages

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, *, in the following expression

```
>> x = 10;
>> 5x
??? 5x
|
Error: Unexpected MATLAB expression.
```

Controlling the hierarchy of operations or precedence

Now we will include ***parentheses***. For example, 4 + 2 * 3 will become (4 + 2) * 3

```
>> (4+2)*3
ans =
18
```

and, from previous example

```
>> 4+2*3
ans =
10
```

By adding parentheses, these two expressions give different results: 10 and 18. The order in which MATLAB performs arithmetic operations is exactly that taught in high school algebra courses. ***Exponentiations*** are done *first*, followed by ***multiplications*** and ***divisions***, and finally by ***additions*** and ***subtractions***. However, the standard order of precedence of arithmetic operations can be changed by inserting *parentheses*. For example, the result of 4+2*3 is quite different than the similar expression with parentheses (4+2)*3.

The results are 10 and 18 respectively.

For operators of *equal* precedence, evaluation is from *left* to *right*. Now, consider another example:

In MATLAB, it becomes $\frac{1}{2+3^2}+\frac{4}{5}\times\frac{6}{7}$

```
>> 1/(2+3^2)+4/5*6/7
ans =
0.7766
```

or, if parentheses are missing,

```
>> 1/2+3^2+4/5*6/7
ans =
10.1857
```

Practice: find the output of the following equations

1. $\boldsymbol{y = 4 - \frac{10}{2+6}}$
2. $\boldsymbol{x = 10, z = \frac{x}{5-3} + \frac{5}{4}}$
3. $\boldsymbol{n = 2, m = 4, t = x^2 + m * \frac{n}{y}}$
4. $\boldsymbol{\frac{z^2-n^2}{z^2-n^2}}$

Lecture Two

Mathematical functions الدوال الرياضية))

There is a long list of mathematical functions that are built into MATLAB. These functions are called built-ins. Examples of these functions are:

sqrt(x)	**square root** الجذر التربيعي	**round(x)**	**Round to nearest integer** التقريب لاقرب عدد صحيح
abs(x)	**absolute value** القيمة المطلقة	**ceil(x)**	**Round toward +∞** تقريب باتجاه اللانهاية الموجبة
sign(x)	**signum function** اشارة القيمة	**floor(x)**	**Round toward -∞** تقريب باتجاه اللانهاية السالبة
real(x)	**Real part of number** الجزء الحقيقي	**log10(x)**	**common logarithm** للوغارثم الشائع
imag(x)	**imaginary part of number** الجزء التخيلي	**cos(x)**	**Cosine** دالة الجيب تمام
mod(x,y)	**Remainder after division** باقي القسمة	**sin(x)**	**Sine** دالة الجيب
exp(x)	**exponential**		

Examples:

```
>> A=5.1;
>> floor(A)
ans =
    5
>> ceil(A)
ans =
    6
>> round(A)
ans =
    5
```

```
>> A= -5.1;
>> sign(A)
ans =
   -1
>> exp(A)
ans =
   0.0061
```

```
>> X=2.3-5.8i;
>> real(X)
ans =
   2.3000
>> imag(X)
ans =
  -5.8000
```

>> A=4; **>> sqrt(A)** ans = 2	**>> A=-5.1;** **>> abs(A)** ans = 5.1000	**>> mod(10,3)** ans = 1

Entering multiple statements per line

It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) to enter more than one statement at once.

يسمح الماتلاب بادخال اكثر من ايعاز ضمن نفس السطر وذلك باستخدام الفارزة (,) او الفارزة المنقوطة (;)

>> **a=7; b=cos(a), c=sin(a)**

b =

0.7539

c =

0.6570

>> C=7.7782-4.9497i, r=real(C),i=imag(C)

C =

7.7782 - 4.9497i

r =

7.7782

i =

-4.9497

How to compute the value of $y = exp^{-a} * \sin(x) + 10\sqrt{y}$ for $a = 5$, $x = 2$, and $y = 8$

>> a = 5; x = 2; y = 8;

>> y = exp(-a)*sin(x)+10*sqrt(y)

y =

28.2904

Homework 1: Find the output of the following MATLAB code:

```
>>sqrt(60);
>>mod(100,5);
>>abs(-10);
>>floor(-20.6);
>>ceil(-20.6)
>>round(-20.6)
```

Matrices (المصفوفات)

What are Matrices?

There are two type of matrices (1 dimension, 2 dimension) matrices

How to write Matrices in MATLAB

❖ **1 dimension**

A=[n1 n2 n3]; Or A=[n1 ,n2 ,n3];

Ex:

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

❖ **2 dimension**

B=[n1 n2 n3;n4 n5 n6];

Ex1:

```
>> B=[1 2 3;4 5 6]
B =
    1    2    3
    4    5    6
```

- يتم كتابة عناصر الصف الأول ,ويتم الفصل بين ارقام الصف الاول اما بوضع فاصلة Comma (,) او مسافة Space
- ويتم الفصل بين عناصر الصف الاول والصف الثاني بوضع الفاصلة المنقوطة Semicolon (;) أو بالضغط على مفتاح Enter
- حجم المصفوفة=عدد الصفوف x عدد الاعمدة

Ex2:

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

Ex3:

```
>> C=[1,2
3,4]
C =
     1     2
     3     4
```

Arithmetic operation on matrices

❖ Addition (+)

$$A=\begin{bmatrix}2 & 1 & 3\\ 3 & 5 & 2\end{bmatrix}_{2x3} \qquad B=\begin{bmatrix}2 & 2 & 1\\ 7 & 9 & 0\end{bmatrix}_{2x3} \qquad A+B=\begin{bmatrix}\mathbf{4} & 3 & 4\\ \mathbf{10} & \mathbf{14} & 2\end{bmatrix}_{2x3}$$

```
>> A=[2 1 3;3 5 2];
>> B=[2 2 1;7 9 0];
>> A+B
ans =
     4     3     4
    10    14     2
```

من شروط عملية الجمع والطرح ان يكون عدد الصفوف في المصفوفة الاولى مساويا لعدد الصفوف في المصفوفة الثانية، وعدد الاعمدة في المصفوفة الاولى يساوي عدد الاعمدة في المصفوفة الثانية. خلافاً لذلك يكون الناتج خطأ

```
>> C=A+B
C =
     4     3     4
    10    14     2
```

❖ Subtraction (-)

$$A=\begin{bmatrix}2 & 1 & 3\\ 3 & 5 & 2\end{bmatrix}_{2x3} \qquad B=\begin{bmatrix}2 & 2 & 1\\ 7 & 9 & 0\end{bmatrix}_{2x3} \qquad A-B=\begin{bmatrix}0 & -1 & 2\\ -4 & -4 & 2\end{bmatrix}_{2x3}$$

```
>> A=[2 1 3;3 5 2];
>> B=[2 2 1;7 9 0];
>> A-B
ans =
    0   -1    2
   -4   -4    2
```

الشرط الاساسي في عملية الضرب ان يكون عدد الاعمدة في المصفوفة الاولى يساوي عدد الصفوف في المصفوفة الثانية. او بالعكس

❖ Multiplication (*)

$$A=\begin{bmatrix}2 & 1 & 3\\ 3 & 5 & 2\end{bmatrix}_{2x3} \qquad B=\begin{bmatrix}2 & 2 & 1\\ 7 & 9 & 0\end{bmatrix}_{2x3} \qquad A*B=\text{Error}$$

$$A=\begin{bmatrix}2 & 1 & 3\\ 3 & 5 & 2\end{bmatrix}_{2x3} \qquad B=\begin{bmatrix}2 & & \\ & 3 & \\ 7 & & 9\\ 0 & 1 & \end{bmatrix}_{3x2} \qquad B*A=\begin{bmatrix}(2*2)+(3*3) & (2*1)+(3*5) & (2*3)+(3*2)\\ (7*2)+(9*3) & (7*1)+(9*5) & (7*3)+(9*2)\\ (0*2)+(1*3) & (0*1)+(1*5) & (0*3)+(1*2)\end{bmatrix}_{3x3}$$

$$A*B=\begin{bmatrix}(2*2)+(1*7)+(3*0) & (2*3)+(1*9)+(3*1)\\ (3*2)+(5*7)+(2*0) & (3*3)+(5*9)+(2*1)\end{bmatrix}$$

```
>> A=[2 1 3;3 5 2]
A =
     2     1     3
     3     5     2
```

```
>> B=[2 3; 7 9;0 1]
    B =
        2     3
        7     9
        0     1
    >> A*B
    ans =
       11    18
       41    56
    >> B*A
    ans =
       13    17    12
       41    52    39
        3     5     2
```

Homework

If A 3x 3 matrices [10 20 30; 1 2 3; 10 20 30], B 2x2 matrices [1 2 3; 11 12 13], find the following: A+B, A-B, A*B, and B*A.

Lecture Three

Vectors (المتجهات)

Is one-dimension matrices either one row or one column.

Creating Vectors: You can create vector in many ways:

You can create a vector by entering each element directly:

1. **Vector_name=[value$_1$ value$_2$ value$_n$]**

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

A =

1 2 3 4 5 6 7 8 9 ← Row vector

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

B =

1
2
3 ← Column vector
4
5
6
7
8
9

Ex:

X=[1 2 3 4 5 6 7 8 9 10]

1 2 3 4 5 6 7 8 9 10

2. Vector_name=[First$_{value}$: Final$_{value}$]

Ex:

Y=[16:20]

16 17 18 19 20

3. Vector_name=[First$_{value}$:increments: Final$_{value}$]

Ex:

Z=[1:2:10]

1 3 5 7 9

4. Vector_name=[Final$_{value}$:Decrement: First$_{value}$]

Ex:

T=[10:-2:1]

10 8 6 4 2

5. linspace

Used to create vector, it creates increased randomized system. Take the following form,

linspace(minimum$_{number}$, maximum$_{number}$, numberof points in between)

A=**linspace**(3,20,6)

لحساب مقدار الزيادة
F= **max$_{num}$ - min$_{num}$**
I=F/(Point-1)

A =

3.0000 6.4000 9.8000 13.2000 16.6000 20.0000

Examples: Write the MATLAB code for creating the following vectors:

- [101 103 105 107 109 111]
- [20 18 16 14 12 10]
- [10 20 30 40 50]
- [130 131 132 133 134 135 136 137 138 139 140]
- [100 75 50 25]

Using linspace create the following vectors

- Vector start with 1 and end with 15 and have 5 points
- Vector start with 15 and end with 100 and have 60 points

Operations on Vectors

No	Name of Operators
1.	**Convert Row vector to Column vector**
2.	**Length of vector**
3.	**Add element**
4.	**Replace element**
5.	**Delete element**
6.	**Call elements**
7.	**Max element**
8.	**Min element**
9.	**Multiplication of elements**
10.	**Summation of elements**

1. Convert Row Vector to Column vector or vice versa (تحويل المتجة الافقي الى المتجه العمودي او العكس)

```
>> A=[1 2 3 4 5 6 7 8 ]
>> A'
ans =
    1
    2
    3
    4
    5
    6
    7
    8
```

2. Length of vector (الطول او عدد العناصر في المتجه)

```
>> Z=[10 3 4 7 9 1 0 4 11]
Z =
   10    3    4    7    9    1    0    4   11
>> length(Z)
ans =
    9
```

3. Add element (اضافة عنصر في نهاية المتجه)


```
>> Z(10)=120
Z =
   10    3    4    7    9    1    0    4   11   120
>> Z(13)=55
Z =
   10    3    4    7    9    1    0    4   11   120    0    0   55
```

- يمكن ان تكون قيم المتجة عبارة عن ارقام متسلسلة, يمكن التعبير عنها باستخدام الامر التالي:

```
>> C=[1:5]
C =
    1    2    3    4    5
```

- هذه الاقواس [] تعبر عن محتوى المتجه اما هذه الاقواس () فهي تعبر عن الموقع وليس القيمة
- فلو اردنا اضافة 5 قيم متسلسلة الى 5 مواقع متسلسلة فنكتب الايعاز التالي

```
>> C(6:10)=[10:14]
C =
    1    2    3    4    5   10   11   12   13   14
```

4. Replace element (استبدال عنصر)

```
>> C(2)=100;
C =
    1   100     3     4     5    10    11    12    13    14
```

- عملية الاستبدال تتضمن تحدد موقع العنصر في المتجهة عبر استخدام (2) ومن ثم وضع القيمه في الموقع المحدد 100.
- يمكن استبدال مجموعة عناصر متتالية عن طريق تحديد بداية الموقع ونهايته كالتالي:

```
>> C(2:5)=[0 0 0 0]
 C =
    1     0     0     0     0    10    11    12    13    14
>> length(C)
ans =
    10
```

5. Delete element(حذف عنصر)

لحذف اي عنصر يجب تحديد موقع العنصر ومن ثم وضع اقواس فارغة [] كالتالي:

لحذف قيمة الموقع الثامن التي هي 12 نكتب الامر التالي:

```
>> C(8)=[]
C =
    1     0     0     0     0    10    11    13    14
>> length(C)
ans =
    9
```

- لحذف مجموعة عناصر متسلسلة نكتب الامر الاتي

```
>> C(1:4)=[]
C =
    0    10    11    13    14
>> length(C)
ans =
    5
```

6. Call elements(استدعاء عنصر)

- يقصد باستدعاءعنصر هو معرفة قيمة العنصر حسب الموقع المحدد فلو اردنا معرفة قيمة الموقع الثالث نكتب التالي:

```
>> C(3)
ans =
   11
```

- ولمعرفة قيم مواقع متسلسلة تكتب الامر الاتي

```
>> C(2:4)
ans =
   10   11   13
```

7. Max element(ايجاد العنصر الاكبر)

- يستخدم الامر max() لايجاد العنصر الاكبر في المتجهة

```
>>C = 0   10   11
  13   14
>> max(C)
ans =
   14
```

8. Min element(ايجاد العنصر الاصغر)

- يستخدم الامر min() لايجاد العنصر الاصغر في المتجهة

```
>> min(C)
ans =
   0
```

9. Multiplication of elements(حاصل ضرب العناصر)

- لايجاد حاصل ضرب عناصر المصفوفة نستخدم ايعاز prod()

```
>> A=[1 2 3 4]
A =
   1   2   3   4
>> B=prod(A)
B =
   24
```

10.Summation of elements (حاصل جمع العناصر)

- الايعاز sum() يتسخدم لايجاد حاصل جمع عناصر المتجهة

>> Z=sum(A)

Z =

10

Homework: If you have the following vector V=[15 30 55 78 9 1 44 100]; Find the following:

1. The length of vector
2. Add [10 20 30], on the positions 9 10 11.
3. Replace the elements from position 4 to position 6 with [1 2 3],
4. Delete the maximum and minimum element.
5. Show the value of second position
6. Find the summation and production of elements.
7. Find the sqrt of each element in the vector

Lecture Four

Operations on the Matrices

No.	Subject	الموضوع
1	Matrices size	حجم المصفوفة
2	Add elements	اضافة عنصر
3	Replace element	استبدال عنصر
4	Delete 1 row or 1 column	حذف صف او عمود
5	Call element	نداء عنصر
6	Call many element	نداء مجموعة عناصر
7	Find max and min element	ايجاد العنصر الاكبر والاصغر
8	Find summation of elements	ايجاد حاصل جمع العناصر
9	Find multiplication of elements	ايجاد حاصل ضرب العناصر
10	Find Diagonal of Matrix	ايجاد القطر الرئيسي في المصفوفة
11	Magic matrix	المصفوفة السحرية
12	Zero matrix	مصفوفة الاصفار
13	One matrix	مصفوفة الواحدات
14	rand matrix	المصفوفة العشوائية

1 Matrices size حجم المصفوفة

To find the size of matrix, we use the code: **size(matrix_name).**

To know the number of rows only: **size(matrix_name,1)**

To know the number of columns only: **size(matrix_name,2)**

Example:

>> A=[3 4 9;2 4 5]	**>> B=[4 2 1; 4 6 7; 1 2 4]**
A =	B =
3 4 9	4 2 1
2 4 5	4 6 7
>> size(A)	1 2 4
ans =	**>> size(B)**
2 3	ans =
>> size(A,1)	3 3
ans =	**>> size(B,1)**
2	ans =
>> size(A,2)	3
ans =	**>> size(B,2)**
3	ans =
	3

2 Add elements اضافة عنصر

For adding new **one** element to matrix, we must write the following statement:

matrix_name(row$_{number}$,column$_{number}$)=value

Example:

>> A=[3 4 9;2 4 5]	**>> B=[4 2 1; 4 6 7; 1 2 4]**
A =	B =
3 4 9	4 2 1
2 4 5	4 6 7
>> A(1,4)=20	1 2 4
A =	**>> B(4,2)=100**
3 4 9 **20**	B =
2 4 5 **0**	4 2 1
	4 6 7
	1 2 4
	0 100 0

For adding new **more** elements to matrix:

Example: suppose we want to add 100,200,300 to the fourth row and first, second, and third column:

>> C=[1 2 3;4 5 6;7 8 9]	**>> Z=[10 ,20,30;11 12 13]**
C =	Z =
1 2 3	10 20 30
4 5 6	11 12 13
7 8 9	
>> C(4,1:3)=[100 200 300]	**>> Z(4,1:3)=[1 2 3]**
C =	Z =
1 2 3	10 20 30
4 5 6	11 12 13
7 8 9	**0 0 0**
100 200 300	**1 2 3**

3 Replace element استبدال عنصر

For replacing one or more elements, we can use the following statement:

matrix_name(row$_{number}$,column$_{number}$)=value

Example:

C =	C =
1 2 3	1 2 3
4 5 6	4 5 0
7 8 9	7 8 9
100 200 300	100 200 300
>> C(2,3)=0	**>> C(1:2,1:3)=0**
C =	C =
1 2 3	**0 0 0**
4 5 **0**	**0 0 0**
7 8 9	7 8 9
100 200 300	100 200 300

4 Delete 1 row or 1 column حذف صف او عمود

We cannot delete one element in the matrix, instead, we can delete complete row or complete column as follow:

Example:

% for delete row	% for delete column
>> F=[3 2 1;6 5 4;9 8 7]	**>> F=[3 2 1;6 5 4;9 8 7]**
F =	F =
3 2 1	3 2 1
6 5 4	6 5 4
9 8 7	9 8 7
>> F(1,:)=[]	**>> F(:,2)=[]**
F =	F =
6 5 4	3 1
9 8 7	6 4
	9 7

5 Call element نداء عنصر

To call one element in matrix: **Matrix_name(row$_{number}$,column$_{number}$)**

Example:

>> F=[3 2 1;6 5 4;9 8 7]	**>> F(3,2)**
F =	ans =
3 2 1	8
6 5 4	**>> F(2,2)**
9 8 7	ans =
>> F(2,3)	5
ans =	
4	

6 Call many element نداء مجموعة عناصر

To call many elements in matrix we must write the follow:

Example:

```
>> B=[1 3 7 8;2 6 5 11;12 14 15 13]
B =
    1    3    7    8
    2    6    5   11
   12   14   15   13
>> B(2,1:4)
ans =
    2    6    5   11
>> B(3,2:4)
ans =
   14   15   13
```

```
>> B(1,end)
ans =
    8
>> B(2,end)
ans =
   11
>> B(end,1)
ans =
   12
>> B(end)
ans =
   13
```

7 Find maximum and minimum element ايجاد العنصر الاكبر والاصغر

To find the maximum or minimum number in matrix, the MATLAB program find the maximum or minimum number in each column by using the following function **max**(**matrix_name**), **min**(**matrix_name**). We need to apply the max or min function twice.

Example:

```
>> F=[1 15 2 11;23 1 4 5;3 1 15 7;1 4 9 10]
F =
    1   15    2   11
   23    1    4    5
    3    1   15    7
    1    4    9   10
```

```
>> N=min(F)
N =
    1    1    2    5
>> b=min(N)
b =
    1
```

```
>> M=max(F)
M =
   23    15    15    11
>> c=max(M)
c =
   23
```

```
>> min(F(1,:))
ans =
    1
>> max(F(3,:))
ans =
   15
```

8 Find summation of elements ايجاد حاصل جمع العناصر

The MATLAB program find the summation of each column in isolation by using **sum(matrix_name)** and put the result in vector**.** Therefore to find the summation of all elements we need to apply **sum()** function twice.

Example:

```
>> F
F =
    1    15     2    11
   23     1     4     5
    3     1    15     7
    1     4     9    10
>> s1=sum(F)
s1 =
   28    21    30    33
>> s2=sum(s1)
s2 =
  112
```

```
>> s3=sum(F(1,:))
s3 =
   29
>> s3=sum(F(3,:))
s3 =
   26
>> s3=sum(F(:,2))
s3 =
   21
```

9 Find multiplication of elements ايجاد حاصل ضرب العناصر

The MATLAB program find the multiplication of each column in isolation by using **prod(matrix_name)** and put the result in vector**.** Therefore to find the multiplication of all elements we need to apply **prod()** function twice.

Example:

```
>> G=[1 2 3;4 5 6;7 8 9]
G =
     1     2     3
     4     5     6
     7     8     9
>> f1=prod(G)
f1 =
    28    80   162
>> f1=prod(f1)
f1 =
      362880
```

```
>> f3=prod(G(1,:))
f3 =
     6
>> f4=prod(G(3,:))
f4 =
   504
>> f5=prod(G(:,2))
f5 =
    80
```

10 Find Diagonal of Matrix ايجاد القطر الرئيسي في المصفوفة

The bellow figure shows the diagonal of matrix, to find it using MATLAB, we use the **diag(matrix_name)** function as follow:

1	15	2	11
23	1	4	5
3	1	15	7
1	4	9	10

Example:

```
>> F=[1 15 2 11;23 1 4 5;3 1 15 7;1 4 9 10]
F =
     1    15     2    11
    23     1     4     5
     3     1    15     7
     1     4     9    10
>> diag(F)
ans =
     1
     1
    15
    10
```

```
>> d=sum(diag(F))
d =
    27
>> d2=prod(diag(F))
d2 =
   150
```

11 Magic matrix المصفوفة السحرية

Is square matrix (number of row=number of column), it contain random numbers MATLAB create magic matrix by using **magic(N)** function.

Example:

>> A=magic(3)	**>> B=magic(5)**
A =	B =
8 1 6	17 24 1 8 15
3 5 7	23 5 7 14 16
4 9 2	4 6 13 20 22
	10 12 19 21 3
	11 18 25 2 9

12 zeros matrix مصفوفة الاصفار

In order to create matriz with zeros element, we can use this function **zeros(m,n)** that creates an mxn matrix whose elements are equal to zero, m is the number of row, and n is the number of column.

Example

>> **zeros(3,4)**	>> **Z=zeros(3,2)**
ans =	Z =
0 0 0 0	0 0
0 0 0 0	0 0
0 0 0 0	0 0

13 ones matrix مصفوفة الواحدات

To creates an mxn matrix whose elements are equal to one, we can use **ones(m,n)** function.

Example

>> **ones(2,3)**	>> **y=ones(3,3)**
ans =	y =
1 1 1	1 1 1
1 1 1	1 1 1
	1 1 1

14 random matrix المصفوفة العشوائية

rand(m,n) creates an mxn matrix whose elements are all random number between 0 and 1

Example:

>> rand(2,4)	>> **x=rand(2,2)**
ans =	x =
0.8147 0.9134 0.2785 0.9649	0.9572 0.1419
0.9058 0.6324 0.5469 0.1576	0.4854 0.4218

Homework

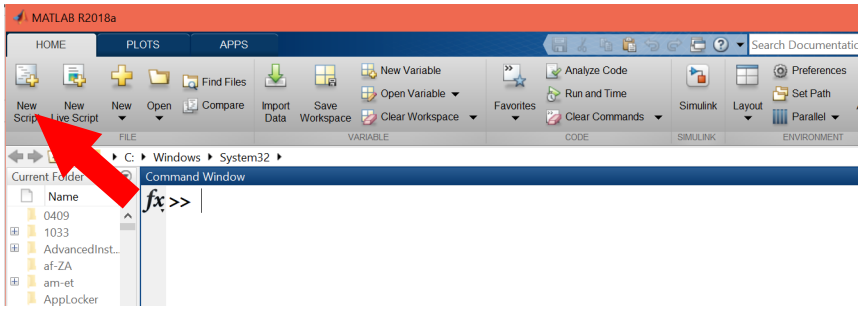
1. Create 3*2 zeros matrix, called A, and replace the position (2,2) with 5.
2. If T= [3 2 1**;** 6 5 4**;** 9 8 7], write the MATLAB codes for the following, with writing the result of operations.
   - (summation of T)-(production of T)
   - (Center of T)*(second position of first row)
   - (Main diagonal elements)*maximum element in T
   - Delete the second row.
3. Create X matrix with size 4 x 3, first and second rows have zeros numbers, while third and fourth rows are ones numbers.
4. If possible, Contact X and T matrix.
5. Create random matrix and find the maximum and minimum number of their diagonal

Lecture Five

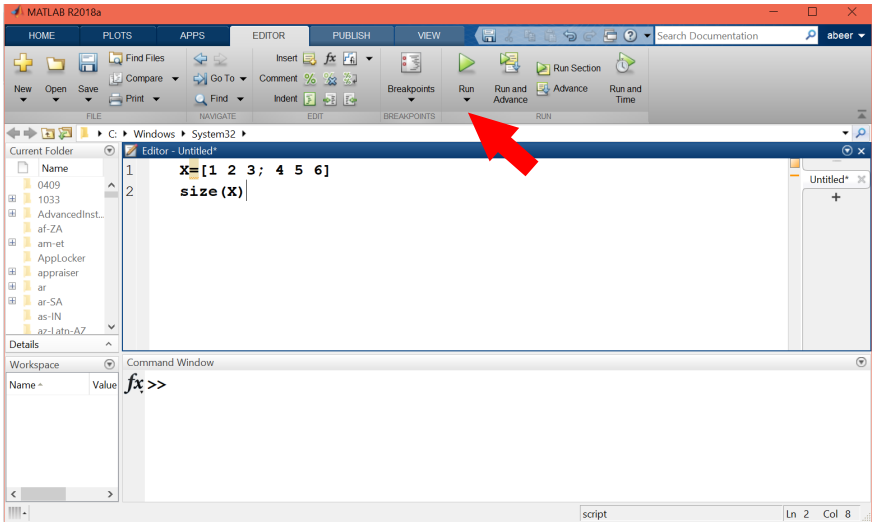
M-Files

How to open file in MATLAB?

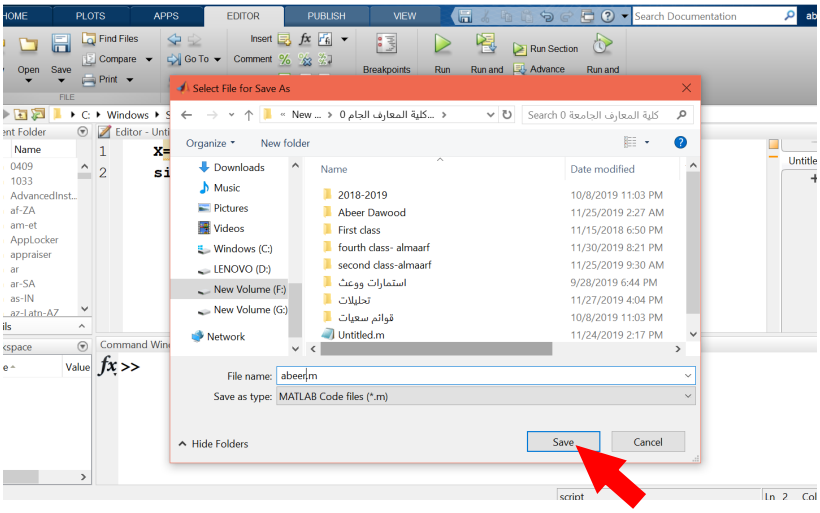
1. Go to Home--New Script as follow



2. Write your MATLAB command in the Script and press on the Run tool



3. After run, MATLAB requires save your file in your favorite location



4. The program will be run and the result will be save in the workspace

﴿Dividing and combining the matrix﴾

If you have X matrix with 3x6 size, Divide it into two matrixes Y and Z with equal size

$$X=\begin{matrix}\mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{2} & \mathbf{0} & \mathbf{9}\\ \mathbf{0} & \mathbf{13} & \mathbf{9} & \mathbf{5} & \mathbf{2} & \mathbf{8}\\ \mathbf{4} & \mathbf{6} & \mathbf{1} & \mathbf{0} & \mathbf{12} & \mathbf{3}\end{matrix}$$

من العمود الاول الى العمود الثالث

>>Y=X(1:3,1:3)

من الصف الاول الى الصف الثالث

$$Y=\begin{matrix}1 & 3 & 5\\ 0 & 13 & 9\\ 4 & 6 & 1\end{matrix}$$

>>Z=X(1:3,4:6)

$$Z=\begin{matrix}2 & 0 & 9\\ 5 & 2 & 8\\ 0 & 12 & 3\end{matrix}$$

Exercise: Divide the following matrix W into four matrixes m, n, f and t with 4x3 size

$$W=\begin{matrix}1 & 2 & 3 & 0 & 1 & 2\\ 0 & 1 & 2 & 1 & 0 & 2\\ 3 & 2 & 1 & 1 & 1 & 0\\ 4 & 3 & 2 & 1 & 2 & 3\\ 2 & 3 & 4 & 3 & 2 & 1\\ 1 & 2 & 3 & 3 & 3 & 2\\ 4 & 3 & 2 & 1 & 2 & 3\\ 1 & 2 & 3 & 1 & 2 & 3\end{matrix}$$

Combine G and H matrices in one matrix R.

G=

10	20
30	40
50	60

H=

70	80
90	100
110	120

>>R=[G H]

R=

10	20	70	80
30	40	90	100
50	60	110	120

Or

>>R=[G;H]

R=

10	20
30	40
50	60
70	80
90	100
110	120

Or

>>R=[G H]'

R=

10	20
30	40
50	60
70	80
90	100
110	120

Exercise: If you have the following matrixes A has zeros elements, B has ones elements, C has random elements, and D has magic elements. Combine its in one big matrix called S as follow.

S=

A	**B**
C	**D**

S=

A	**B**	**C**	**D**

Lecture Six

2D Plotting

2D Plot is the relation between two Variables:

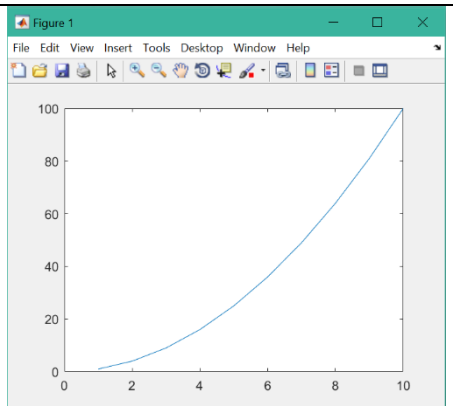
1. ***Independent Variable:*** Values entered by user don't have any relation with anyone.
2. ***Dependent Variable:*** Values can be found by applying the relation with Independent Variable

plot(Independent Variable, Dependent Variable)

EX:

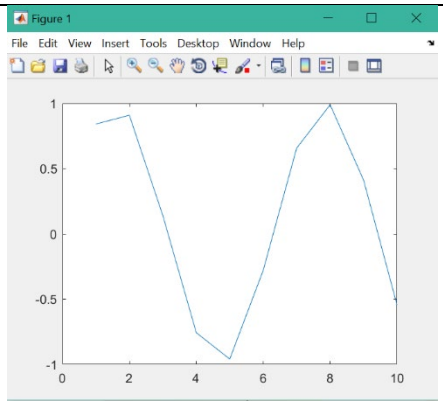
```
clc
clear
X=1:10
Y=X.^2
plot(X,Y)
```

```
X =
   1   2   3   4   5   6   7   8   9  10
Y =
   1   4   9  16  25  36  49  64  81 100
```



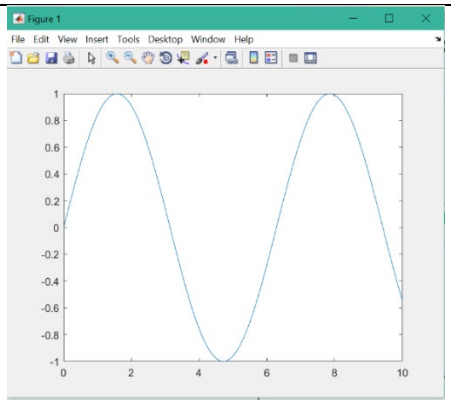

```
clc
clear
X=1:10
Y=sin(X)
plot(X,Y)
```

```
X =
   1 2 3 4 5 6 7 8 9 10
Y =
  0.8415   0.9093   0.1411  -0.7568  -0.9589  -
0.2794  0.6570 0.9894 0.4121
  -0.5440
```



```
clc
clear
X=0:0.1:10
Y=sin(X)
plot(X,Y)
```

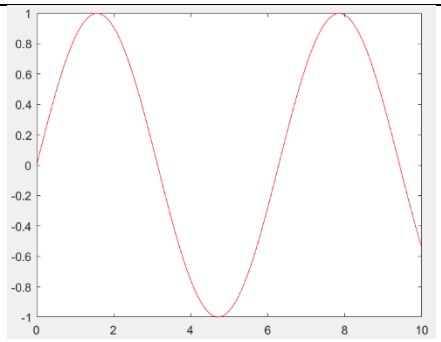
لجعل القيم اكثر دقة تؤخذ كل الارقام بين 0و1 حتى القيم العشرية



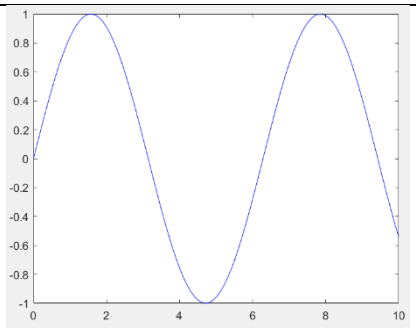
Plot Proprieties

plot(Independent Variable, Dependent Variable, 'the property')

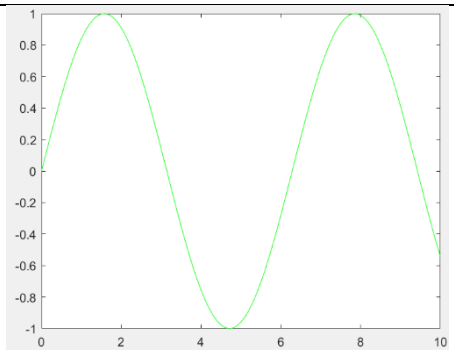
No.	Subject	Color	MATLB commend
1	b	Blue	plot(X,Y,'b')
2	g	Green	plot(X,Y,'g')
3	r	Red	plot(X,Y,'r')
4	m	Magenta	plot(X,Y,'m')
5	k	black	plot(X,Y,'k')
6	+	plus	plot(X,Y,'+')
7	*	Star	plot(X,Y,'*')
8	s	square	plot(X,Y,'s')
9	r*	red&star	plot(X,Y,'r*')
10	grid		
11	Hold on hold off		
12	figure		
13	subplot		
14	xlable & ylable		
15	Legend		



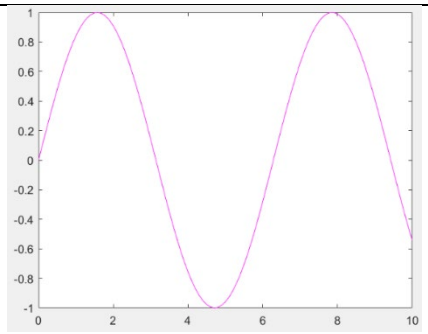
plot(X,Y,'r')



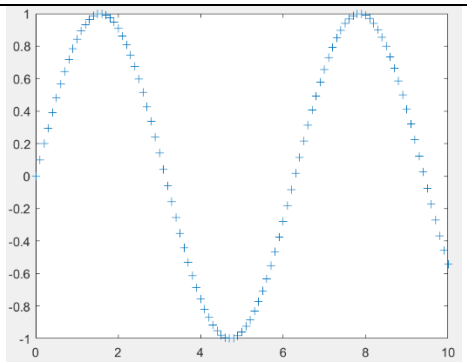
plot(X,Y,'b')



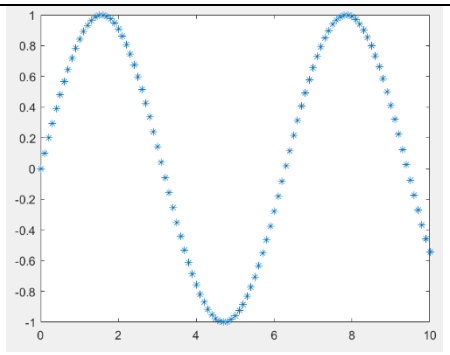
plot(X,Y,'g')



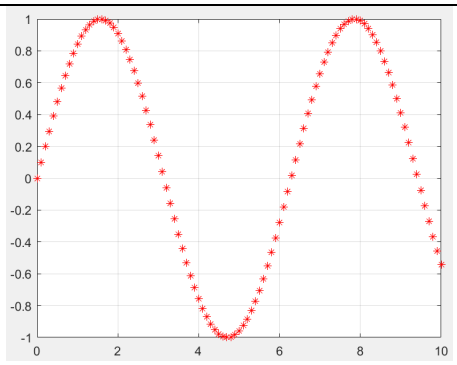
plot(X,Y,'m')



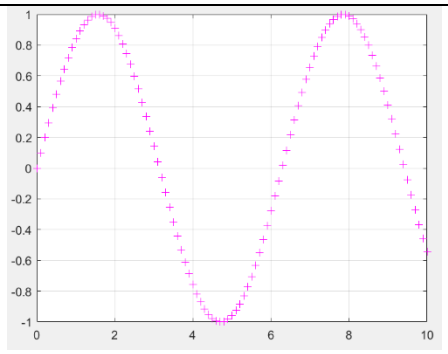
plot(X,Y,'b+')



plot(X,Y,'b*')



plot(X,Y,'r*')



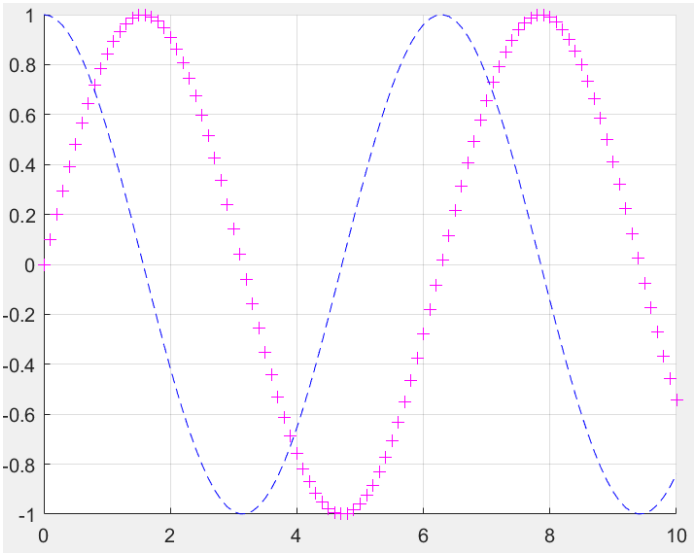
plot(X,Y,'m+')

Plot more than one plot in the same figure, **hold on**, and **hold off** used for that purpose. It placed before the plot commend.

- رسم اكثر من دالة داخل شكل واحد

Ex:

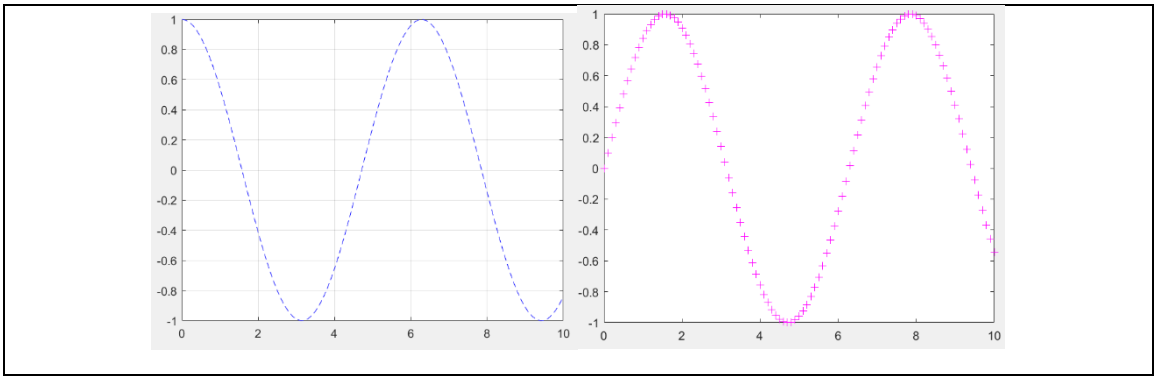
```
clc
clear
X=0:0.1:10
Y=sin(X)
Z=cos(X)
hold on
plot(X,Y,'+m')
plot(X,Z,'b--')
hold off
grid
```



To place every plot in isolated figure we can use **figure**, before the plot commend

- رسم كل دالة داخل شكل خاص بها

```
clc
clear
X=0:0.1:10
Y=sin(X)
Z=cos(X)
plot(X,Y,'+m')
figure, plot(X,Z,'b--')
grid
```



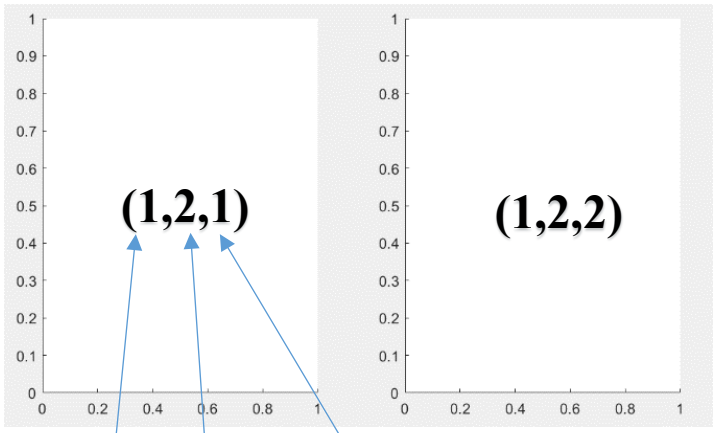
Place isolated plots in one Window, by using **subplot**. That makes the figures as matrix or vector. You must know the number and position of figures.

subplot(*number of row, number of column, the number of figure which occupy the figure***)**

- **رسم اكثر من دالة على شكل خاص بها لكن داخل نفس النافذة**

subplot(1,2,1)

subplot(1,2,2)



subplot(2,3,1)

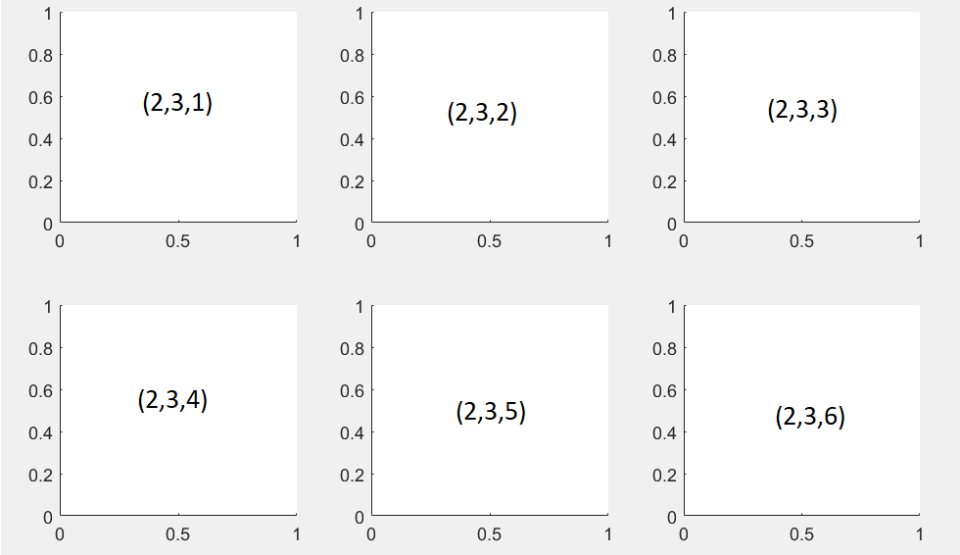
subplot(2,3,2)

subplot(2,3,3)

subplot(2,3,4)

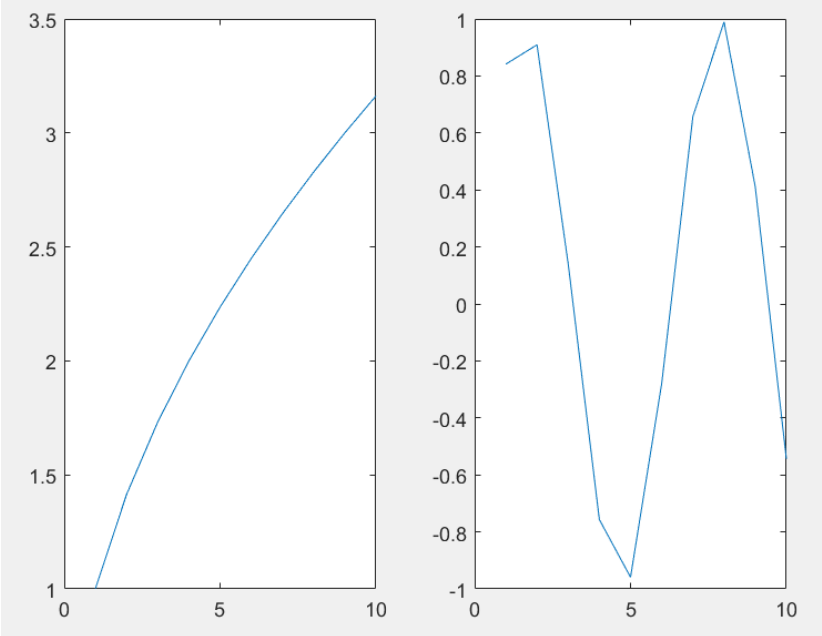
عدد الصفوف

عدد رقم الخانة التي يشغلها



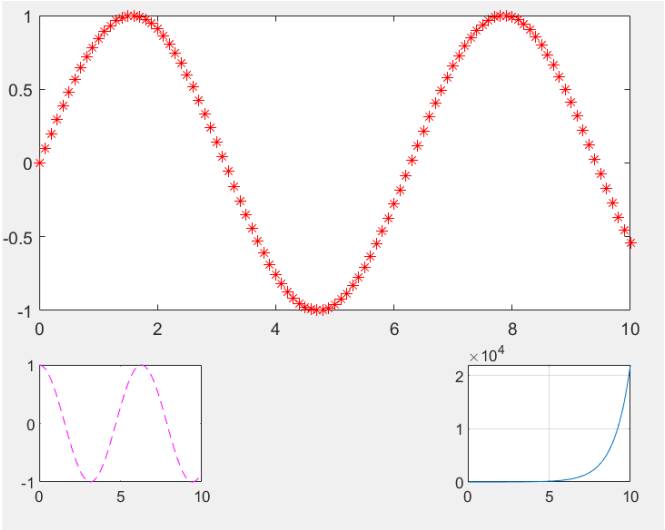
EX:

```
clc
clear
X=[1:10]
Y=sqrt(X)
Z=sin(X)
subplot(1,2,1),plot(X,Y)
subplot(1,2,2),plot(X,Z)
```



Example:

```
clc
clear
X=0:0.1:10
Y=sin(X)
Z=cos(X)
V=exp(X)
subplot(3,3,[1 2 3 4 5 6])
plot(X,Y,'*r')
subplot(3,3,7)
plot(X,Z,'m--')
subplot(3,3,9)
plot(X,V)
grid
```

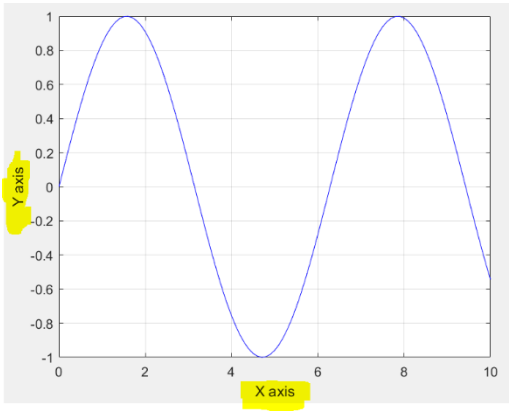


Axis Naming **تسمية المحاور**

For naming the x-Axis and y-Axis, we can use the following command:

```
xlabel(' The name of axis ')
ylabel(' The name of axis ')
```

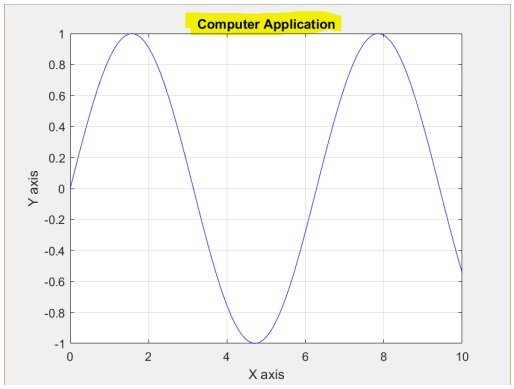
```
clc
clear
x=0:0.1:10;
y=sin(x)
plot(x,y,'b')
xlabel(' X axis ')
ylabel(' Y axis ')
grid
```



Title the plot وضع عنوان اعلى الشكل

يمكن وضع عنوان أعلى آلرسمة من وذلك خلال الأمر (title) حيث هذا الأمر بالشكل التالي

```
title('The title of the Figure ')
```

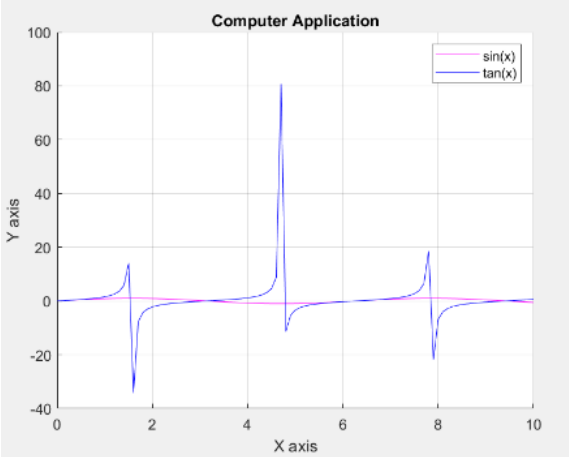


Legend

يستخدم لوضع دليل في اعلى الرسم ليوضح ماذا يعني كل لون داخل الرسم. ويكتب بهذا الشكل.

```
Legend('The color reference')
clc
clear
x=0:0.1:10;
y=sin(x)
z=tan(x)
hold on
plot(x,y,'m');
plot(x,z,'b');
```

```
legend('sin(x)','tan(x)');
hold off
xlabel(' X axis ')
ylabel(' Y axis ')
title('Computer Application')
grid
```



Lecture Seven

Input and Output statements in MATLAB

Input Statement

لكي يقوم المستخدم بادخال النصوص والارقام الى الماتلاب عن طريق command widow نستخدم دالة input

لادخال رقم نستخدم دالة input بالشكل التالي:

```
A=input(' Enter num ')
```

Write code in the command window and press enter

```
>> A=input(' Enter num ')
 Enter num
```

Enter a number. For example 6 and press enter

```
>> A=input(' Enter num ')
 Enter num 6
```

A will have the value that you are entered

```
>> A=input(' Enter num ')
 Enter num 6

A =

     6
```

For input characters or string, write the following format:

لادخال الحروف,نقوم بوضع حرف ' s ' داخل دالة input ليكون بهذا الشكل:

```
A=input(' Enter string ','s')
```

Write code and press enter, and then enter any string finally press enter as follow:

```
>> A=input('Enter charecters','s')
Enter charectersabcd

A =

    'abcd'
```

Example: enter your name and age (write code in the script)

```
clear
clc
A=input('Enter your name','s');
B=input('Enter your age');
```

```
Enter your nameahmed

A =

    'ahmed'

Enter your age22

B =

    22
```

- لمعرفة نوع البيانات الي ادخلت نستخدم دالة class

```
>> class(A),class(B)

ans =

    'char'


ans =

    'double'
```

لو قمنا بكتابة الكود التالي في script لادخال رقمين وايجاد حاصل جمعهما بالشكل التالي

A=input('Enter num1')

B=input('Enter num2')

C=A+B

في command window سوف يطلب ادخال الارقام وعند الضغط على مفتاح Enter سوف يطبع النتاج المخزن في المتغير C

Enter num15

A =

5

Enter num26

B =

6

C =

11

الان, لو قمنا باعادة كتابة نفس الكود لكن نضع حرف s على احد المتغيرين او كلاهما

```
clc
clear
A=input('Enter num1','s')
B=input('Enter num2')
C=A+B
```

فأن البرنامج سوف لن يقوم بعملية الجمع بالشكل الصحيح لان المتغير الاول هو حرف وليس رقم لهذا البرنامج سوف يأخذ القمية الرقمية للحرف المدخل حتى لو قمنها بكتابه 5

Enter num1 **5**

A =

'5'

Enter num2 **6**

B =

6

C =

59

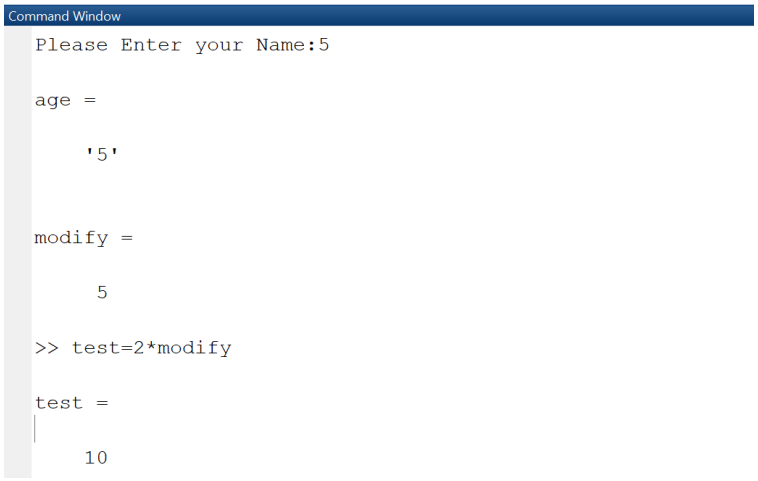
تحويل من نص الى رقم ومن رقم الى نص **str2num(), num2str()**

للتحويل من حرف الى رقم نستخدم الدالة **str2num()**

بالمثال ادناه قمنا بادخال العمر بشكل حرف عن طريق كتابة حرف s لتحويل العمر من حرف الى رقم استخدمنا الدالة **str2num ()**

```
clear
clc
age=input('Please Enter your Name:','s')
modify=str2num(age)

```



لتحويل الرقم الى حرف نستخدم **num2str()**

في المثال ادناه قمنا بادخال الاسم نوع حرف وقمنا بادخال العمر بشكل رقم ونريد تخزين الاسم والعمر بمتجه. وكما نعرف ان عناصر المتجهات يجب ان تكون من نوع بيانات واحد اما جميع العناصر عبارة حروف واما جميعها ارقام. لو فرضنا ان جميع العناصر هي حروف فيجب تحويل العمر من رقم الى حرف عن طريق دالة **num2str**

Script

```
clc
clear
A=input('Enter name','s')
B=input('Enter age')
std=[A, num2str(B)]
```

Command window

```
Enter namemohammed
A =
   'mohammed'
Enter age22
B =
   22
std =
   'mohammed22'
```

Example1: Read number	**Example2: Read string**
clear; clc; X=input('X= ') Y=input('Y= ') Z=X+Y	clear; clc; X=input('X= ','s') Y=input('Y= ','s') Z=X+Y
output	**output**
X= 10 X = 10 Y= 5 Y = 5 Z = 15	X= 10 X = '10' Y= 5 Y = '5' Z = 102 101
Solving the problem	**Example: Read number and string**
clear; clc; X=input('X= ','s') Y=input('Y= ','s') Z=str2num(X)+str2num(Y)	clear; clc; name=input('Name: ','s') stage=input('stage: ') info=[name ,num2str(stage)]
output	**output**
X= 10 X = '10' Y= 5 Y = '5' Z = 15	Name: abeer name = 'abeer' stage: 2 stage = 2 info = 'abeer fallujah 2'

خلاصة الدوال

Input(' ')	دالة ادخال الارقام
Input(' ','s')	دالة ادخال الحروف
class	لمعرفة نوع البيانات للمتغير
str2num()	دالة تحويل الحرف الى رقم
num2str()	دالة تحويل الرقم الى نص

Homework:

1. **Write MATLAB code to enter your (name, age, address, college, department, and stage).**
2. **Show the class of each input data.**
3. **Collect your information into one std vector that has class cha**

Lecture Eight

Input and Output statement in MATLAB

Output functions

There are three common ways for printing in MATLAB:

- Type the name of a variable without a trailing semi-colon.
- Use the "**disp**" function.
- Use the "**fprintf**" function, which accepts a C printf-style formatting string**.**

للطباعة داخل الماتلاب يمكن استخدام 3 اساليب وهي

- كتابة المتغير بدون وضع فارزة منقوطة نهاية الايعاز
- استخدام دالة disp()
- استخدام دالة(fprintf التي تحتوي تنسيقات تشبه تنسيقات لغة السي

Here are examples:

```
>> x=[1 2 3 4];
>> x
x =
    1    2    3    4
>> disp(x)
    1    2    3    4
>> fprintf('%i\n',x)
1
2
3
4
```

Note:

"**disp**" excludes the variable name from the output

تكتب دالة disp بالشكل التالي:

disp(variable)

"**fprintf**" uses the formatting string on each element of the variable. In the example above, applying "\n" to each element had the net effect of printing a row vector as if it were a column vector.

fprintf('%i\n',x)

هذه الدالة تقوم بالطباعة ضمن تنسيقات معينة حيث يجب وضع علامة % ومن ثم نوع المتغير اذا كان double, integer نضع حرف i, اما اذا كان المتغير عبارة عن حروف نضع اما حرف c,s بعد علامة %

لوضع سطر جديد نستخدم %i\n , لوضع مسافة بين المتغيرات نستخدم %i\t, اما لوضع \ بين المتغيرات نستخدم %i\\, واخيرا لوضع علامة النسبة المئوية لمتغير ما نستخدم %i%%.

Value Type	Conversion	Details
Integer, signed	%d or %i	Double or Integer number
Characters or strings	%c	Single character
	%s	Character vector or string array.

Examples

The command fprintf('Hello');

Displays the text 'Hello' on the icon. The command fprintf('Hello = %d',17);

Uses the decimal notation format (%d) to display the variable 17.

Character	Description	Example	
\n	Newline	fprintf('%\n',17)	17
\t	Horizontal tab	fprintf('%\t',17)	17
\\	Backslash	fprintf('%\\',17)	17\
%%	Percent character	fprintf(' %%%',17)	17%

Example

```
A=10;
B=20;
C=A+B;
disp(A)
disp(B)
disp(C)
```

Output in command window:

```
10
20
30
```

```
A=10;
B=20;
C=A+B;
fprintf('%d\n%d\n%d\n',A,B,C);
```

Output in command window:

```
10
20
30
```

```
A=10;
B=20;
C=A+B;
fprintf('%d\\%d\\%d\\',A,B,C);
```

Output in command window:

```
10\20\30\
```

```
A=10;
B=20;
C=A+B;
fprintf('%d%%%d%%%d%%',A,B,C);
```

Output in command window:

```
10%20%d%30%
```

```
A=10;
B=20;
C=A+B;
fprintf('%d\t%d\t%d\t',A,B,C);
```

Output in command window:

```
10      20      30
```

Homework

1. **Write MATLAB code to enter your (name, age, address, college, department, and stage).**
2. **Write MATLAB code to Print the information each one in new line**
3. **Print the information with \ between the input data**
4. **Print the information with space between the input data**
5. **Print the information with % between the input data**

Lecture Nine

Conditional and loop Statements

Relational operators

Relational operator	Description	Example	Result
<	less than	2 < 3	1
<=	less than or equal to	2 <= 3	1
>	greater than	2 > 3	0
>=	greater than or equal to	2 >= 3	0
==	equal to	2 == 3	0
~=	not equal to	2 ~= 3	1

Conditional Statement (If-elseif-else)

If-else

If (condition) <statements> **end**	**If** (condition) <statements> **else** <statements> **end**

Example1
a = 10;
if (a < 20)
fprintf('value of a: %d\n', a);
end
 	
Output:
value of a: 10
	Example2
	a = 60;
	if (a > 50)
	disp('pass');
	else
	disp('fail');
	end
	Output:
	pass

Example 3 Calculate the square root y of the variable x only when the value of x is non-negative.

```
x=input('enter number: ')
if x>=0
y=sqrt(x)
end
```

Example 4: Give MATLAB code to calculate y where y = -1 when x < 0 and y = 2 when x > 2.

```
x=input('enter number: ')
 if x<0
        y=-1
 elseif   x>2
        y=2
end
```

Example 5: The value of f(x) is -2x when x < 0; x(x-2) when x is in [0, 2] and log(x-1) otherwise. Calculate f(x).

```
x=input('enter number: ')
if x<0
    f=-2*x
elseif x<=2
    f=x*(x-2)
else
    f=log(x-1)
 end
```

Example 6: Write MATLAB program to enter employ name, age, and salary, and test if the age greater than 50 increase the salary by 25000 and print the information with putting newline after each info.

```
name=input('employ  name:','s');
age=input('employ  age:');
```

```
salary=input('employ  salary:');
 if (age>50)
salary=salary+25000;
 end
fprintf('%s\n%d\n%d\n,name,age, salary);
```

Homework

1. Write program to input number and test if it positive or negative using MATLAB.
2. Write program to input number and test if it even or odd using MATLAB.
3. Write program to input name and test if it 'ahmed' print 'male' or 'noor' print 'female' using MATLAB.
4. Write program to input number and test if it not equal 10 then calculate their sin function.

Loop statements (For and while)

For

for index = Start count :End count

<program statements>

...

end

Example 1	**Output**
for a = 10:15 fprintf('value of a: %d\n', a); **end**	value of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 15

Example 2	**Output**
for a = 1.0: -0.1: 0.0 disp(a) **end**	1 0.9000 0.8000 0.7000 0.6000 0.5000 0.4000 0.3000 0.2000 0.1000 0

Example 3: Given an integer n, calculate the square of the integers k^2 for k = 1,...,n. store the results in vector s.

```
n=5;
for k=1:5
   s(k)=k*k
end
```

Output

```
s =
    1    4    9   16   25
```

***Example 4*:** Program to enter 10 elements and save it in vector X and print it each one on new line

```
for i=1:10
    X(i)=input('Enter number');
End
for i=1:10
    fprintf('%i\n', X(i));
end
```

***Example 5*:** Write program to enter 10 numbers, put it in vector Y (using for) and print the maximum and minimum number.

```
for i=1:10
   s(i)=input('Enter num')
end
    fprintf('The max number is: %i\\', max(s));
    fprintf('The min number is: %i\\', min(s));
```

Switch-case

```
variable
switch switch_expression
  case case_expression
    <statements>
  case case_expression
    <statements>
   ...
  otherwise
    <statements>
end
```

Example 1:

```
n = input('Enter a number: ');
switch n
   case -1
      disp('negative one')
   case 0
      disp('zero')
   case 1
      disp('positive one')
   otherwise
      disp('other value')
end
```

Example2: Write MATLAB program to enter student's average and check the cases if average =90 print `'Excelent'`, if average =80 print `'Very Good'`, if average =70 print `'Good'`, if average =60 print

'Intermediate', if average =50 print 'pass' otherwise print 'fail' use 'switch-case'.

```
    s=input('input average:')
  switch s
    case 90
      disp('Excelent')
    case 80
      disp('Vary Good')
    case 70
       disp('Good')
    case 60
       disp('Intermediate')
    case 50
       disp('pass')
    otherwise
      disp('fail')

end
```

Homework

1. **Write program to enter 10 numbers, put it in vector Y and print the production and summation of all elements.**
2. **Write program to enter 7 student's marks, and find the average of its marks.**
3. **Write program to enter 2 numbers, and swap between them.**
4. **Write MATLAB program to enter week day and check the cases if day ='1' print 'Saturday', if day =2 print 'Sunday', if day =3 print 'Monday', otherwise print 'other' use 'switch-case'.**

Lecture Ten

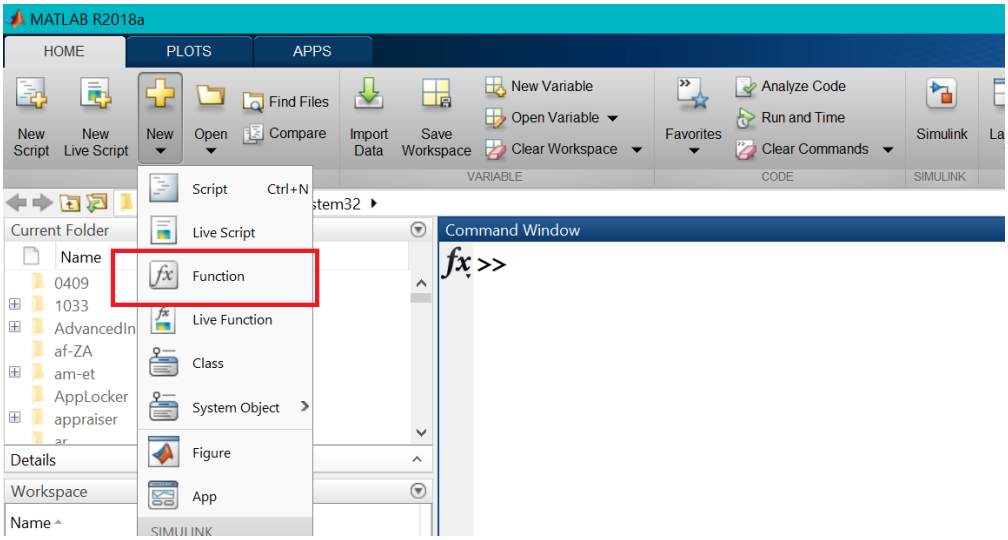
Functions in MATLAB

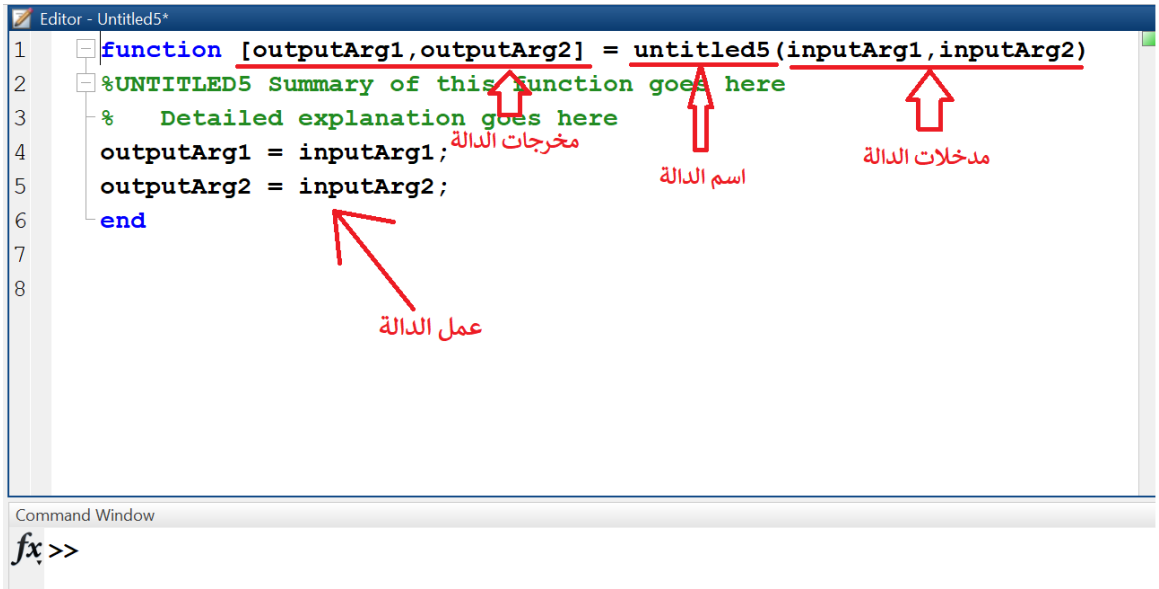
Create Function

There are two ways to create the function in MATLAB as the following:

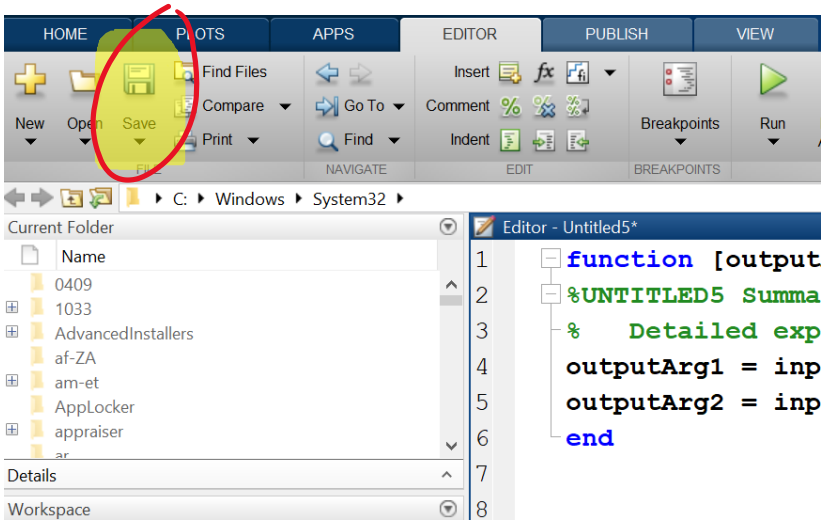
First way

للإنشاء دالة اتبع الخطوات ادناه:

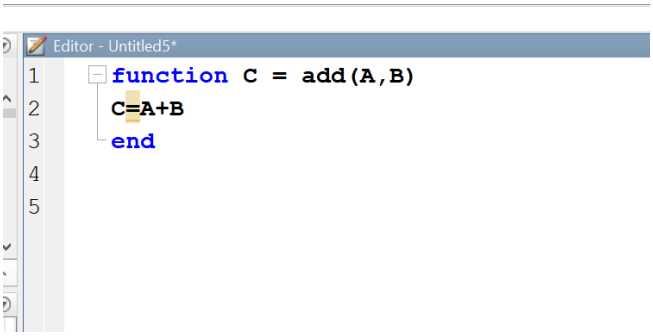


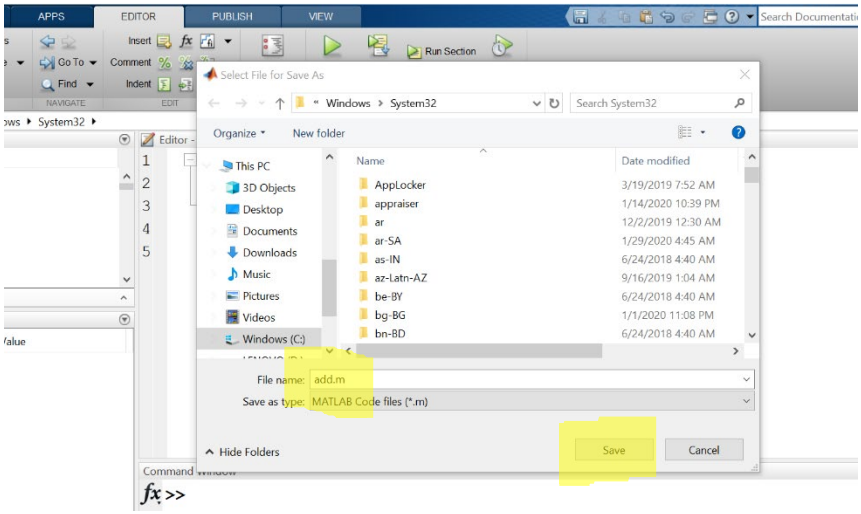


بعد الانتهاء من عمل الدالة يجب حفظ الدالة كما في الصورة ادناة



Ex:





Second way

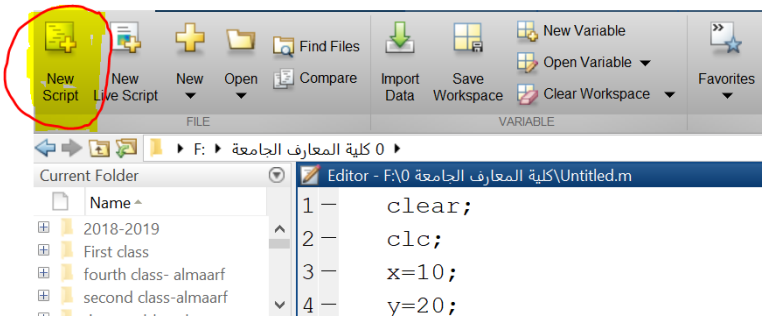
1. **Write the program in the main script.**

نقوم بفتح script من اجل كتابة البرنامج الرئيسي

2. **Open a new script for writing the function code.**

نقوم بفتح script اخر من اجل كتابة الدالة

As follow:



3. Inside the function script, write the following:

كاتبة الدالة تكون بهذا الشكل

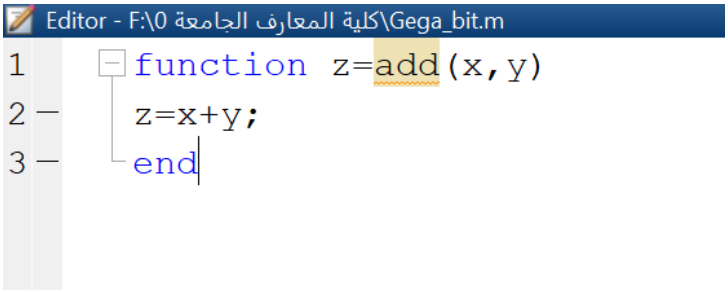
المخرجات اسم الدالة المدخلات

```
Function outputs=function_name(inputs)
```

……..

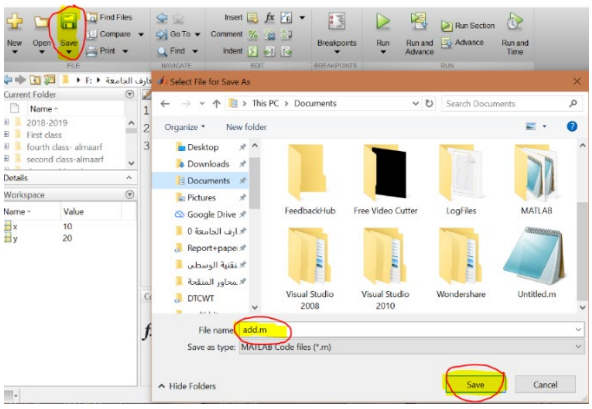
……..

End



4. Save the function as follow:

بعدها نقوم بحفظ الدالة كما موضح في الصورة



Call Function

A function called inside the main program as follow:

استدعاء الدالة داخل البرنامج الرئيسي وكالتالي

outputs=function_name(inputs)

```
clear;
clc;
x=10;
y=20;
z=add(x,y);
fprintf('%i\n',z);
```

Examples:

1. Write a program to enter two variables and calculate the four maths operations on it using the **function**.

```
function [a,s,m,d]=math(x,y)
a=x+y;
s=x-y;
m=x*y;
d=x/y;
end
```

```
clear;
clc;
x=input('Enter num1');
y=input('Enter num2');
[a,s,m,d]=math(x,y);
fprintf('%i\n%i\n%i\n%f\n',a,s,m,d);
```

2. Write a program to enter 10 numbers and find the summation, production, maximum, and minimum of it inside **the function**.

```
function [s,p,m,n]=math(x)
s=sum(x);
p=prod(x);
m=max(x);
n=min(x);
end
```

```
clear;
clc;
for i=1:10
   x(i)=input('enter num')
end
[s,p,m,n]=math(x);
fprintf('sum=%i\n prod:%i\n max:%i\n min:%i\n',s,p,m,n);
```

3. Write a program to enter the storage space in the bit unit and convert it to Gega Byte using a function called **bit_Gega** and vice versa using another function called **Gega_bit**

```
function Gega=bit_Gega(bit)
Byte=bit/8;
Kelo=Byte/1024;
Mega=Kelo /1024;
Gega=Mega/1024;
end
```

```
function b=Gega_bit(Gega)
mage=Gega*1024;
kelo=mega*1024
byte= kelo *1024;
b=byte*8;
end
```

```
clear;
clc;
bit=input('Enter the storage space:');
Gege=bit_Gega(bit);

fprintf('The storage in bit=%i\n The storage in Gega
byte:%f\n',bit,Gege);

bit2=Gega_bit(Gege)
fprintf('The storage in bit again=',bit2)
```

Output:

```
Enter the storage space:123456788
The storage in bit=123456788
 The storage in GegeByte:14.717196
The storage in bit again=123456788
```

Homework: Write a program to enter the time in hours and convert it to second using **function** called **Hour_ Seconds** and vice versa using another function called **Seconds_ Hour.**

Good Luck