

# المحاكاة بالحاسب الآلي برنامج الماتلاب

## جدول المحتويات

٦	الفصل الأول نظرة عامة على الماتلاب
٧	مقدمة
٧	تعريف برنامج الماتلاب
٧	مكونات شاشة الماتلاب
٨	نافذة الأوامر Command Window
٨	منطقة العمل Workspace
٨	نافذة تسجيل الأوامر Command History
٨	قائمة ملف File
٩	قائمة التعديل Edit
٩	قائمة Debug
٩	قائمة Desktop:
١٠	قائمة Window
١٠	قائمة Help
١٠	العمليات الحسابية
١١	عملية وضع الأس
١٢	أخذ الجذر التربيعي sqrt
١٢	وضع تعليقات أثناء البرمجة
١٢	تعريف المتغير
١٣	ملاحظات على المتغيرات
١٥	الفاصلة المنقوطة نهاية الجملة
١٥	إلغاء القيم المدخلة والنتائج
١٦	عملية المسح الجزئي للمتغيرات
١٧	الأوامر الخاصة ببرنامج الماتلاب
١٧	الدوال المثلثية Trigonometric functions
١٨	الدوال المثلثية العكسية Inverse Trigonometric functions
١٨	الدالة الأسية Exponential Function
١٩	الجذر التربيعي sqrt
١٩	الأعداد المركبة Complex Numbers
١٩	الكتابة فوق قيمة العدد المركب

٢١	..... Natural Logarithm اللوغاريتمات الطبيعية
٢١	..... العمليات التقريبية لأعداد واقعة بين رقمين
٢٤	..... Matrices الفصل الثاني المصفوفات
٢٥	..... تعريف المصفوفة
٢٥	..... كيفية كتابة المصفوفات في برنامج الماتلاب
٢٦	..... العمليات الأساسية التي تتم على المصفوفات
٢٦	..... الجمع والطرح
٢٨	..... ضرب المصفوفات
٢٩	..... ضرب المصفوفة عنصر لعنصر
٣٠	..... قسمة المصفوفات
٣٢	..... الفرق بين المتجهات والمصفوفات
٣٣	..... العمليات الشائعة على المتجهات
٣٣	..... طول المتجه
٣٤	..... إضافة عنصر لمتجه
٣٥	..... إضافة أكثر من عنصر متتالي لمتجه
٣٦	..... استبدال عنصر في متجه
٣٧	..... استبدال مجموعة عناصر متتالية في متجه
٣٨	..... حذف عنصر من المتجه
٣٩	..... حذف مجموعة عناصر متتالية في متجه
٣٩	..... نداء عنصر من متجه
٤٠	..... نداء أكثر من عنصر من متجه
٤١	..... إيجاد العنصر الأكبر في المتجه باستخدام الأمر max
٤٢	..... إيجاد العنصر الأصغر في المتجه باستخدام الأمر min
٤٢	..... إيجاد مجموع عناصر المتجه باستخدام الأمر sum
٤٢	..... إيجاد حاصل ضرب العناصر في المتجه باستخدام الأمر prod
٤٢	..... إنشاء متجه باستخدام الأمر linspace
٤٣	..... إنشاء متجه باستخدام الأمر randint
٤٤	..... العمليات الشائعة على المصفوفات
٤٤	..... إيجاد حجم المصفوفة باستخدام الأمر size
٤٥	..... إضافة عنصر إلى المصفوفة
٤٦	..... استبدال عنصر في مصفوفة

٤٧	حذف أكثر من عنصر من مصفوفة.....
٤٧	نداء عنصر من مصفوفة.....
٤٨	إيجاد العنصر الأكبر في مصفوفة باستخدام الأمر max.....
٤٩	إيجاد العنصر الأصغر في مصفوفة باستخدام الأمر min.....
٤٩	إيجاد مجموع العناصر في مصفوفة باستخدام الأمر sum.....
٥٠	إيجاد حاصل ضرب العناصر في مصفوفة باستخدام الأمر prod.....
٥١	إيجاد قطر المصفوفة باستخدام الأمر diag.....
٥٢	إنشاء المصفوفة السحرية باستخدام الأمر magic.....
٥٣	إيجاد جذور المصفوفة باستخدام الأمر roots.....
٥٣	إنشاء مصفوفة ذات قيم تسلسلية.....
٥٥	الفصل الثالث الملفات M-File.....
٥٦	تعريف M-File.....
٥٩	الأمران CLC و Clear.....
٦٢	الأمر Input.....
٦٢	الأمر disp.....
٦٣	التحويل بين string و number.....
٦٦	الفصل الرابع الجمل الشرطية والبرمجة.....
٦٧	المقارنة.....
٦٨	الأمر /n.....
٦٩	الأمر If.....
٦٩	.....simple if statement
٦٩	.....nested if statement
٦٩	.....if-else statement
٧٢	الأمر For.....
٧٣	الأمرين Break و Continue.....
٧٥	الأمر While.....
٧٦	الفصل الخامس الرسم.....
٧٧	الرسم ثنائي الأبعاد 2D Plotting.....
٧٧	الأمر Plot.....
٧٩	عملية وضع شبكة على الرسم.....

٧٩	إضافة خصائص إلى الرسومات .....
٨٣	وضع الرسومات في نوافذ منفصلة .....
٨٥	إنشاء رسومات منفصلة في نافذة واحدة .....
٩٠	تسمية المحاور .....
٩٠	وضع عنوان في أعلى الرسم .....
٩١	وضع نص على نقطة أو أكثر داخل الرسم .....
٩٧	فتح نافذة جديدة وتحديد دقتها .....
٩٧	مثال تطبيقي .....
٩٨	كيفية إدخال النقاط من خلال الماوس .....
٩٩	الرسم ثلاثي الأبعاد .....

# الفصل الأول

## نظرة عامة على الماتلاب

## مقدمة

يعتبر برنامج الماتلاب اليد اليمنى لأي مهندس مهما كان مجاله، نظراً لما يحتويه من مرونة في التعامل مع العمليات الحسابية فهو يوفر الوقت والجهد والدراية. سيتم تقسيم تعليم برنامج الماتلاب إلى أقسام، تبعاً لنوع التطبيق.

موقع الشركة المصنعة لبرنامج الماتلاب

<http://www.mathworks.com>

ستجد في هذا الموقع آخر الإصدارات لبرنامج الماتلاب، وكذلك التحديثات الخاصة بالبرنامج، كما يوفر شرحاً (باللغة الإنجليزية) لبرنامج الماتلاب.

## تعريف برنامج الماتلاب

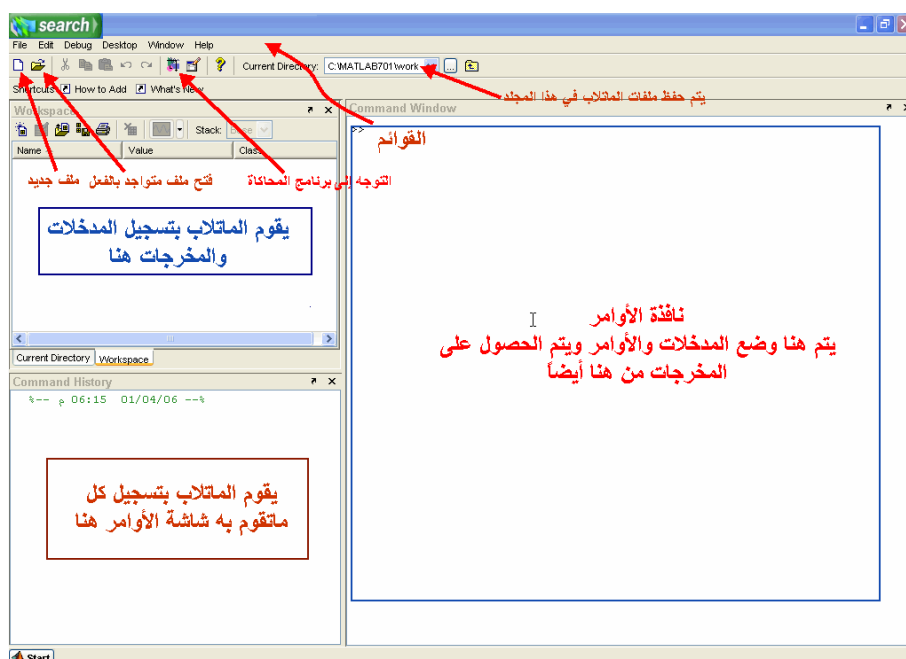
برنامج الماتلاب هو برنامج هندسي (وله مجالات أخرى) يقوم بعمليات تحليل وتمثيل البيانات من خلال معالجة تلك البيانات تبعاً لقاعدة البيانات الخاصة به، فمثلاً يستطيع البرنامج عمل التفاضل differentiation والتكامل Integration وكذلك يقوم بحل المعادلات الجبرية Algebraic Equations ذات الرتب العليا والتي قد تصل من الصعوبة ما تصل، ليس ذلك فقط بل يستطيع البرنامج عمل التفاضل الجزئي، ويقوم بعمل عمليات الكسر الجزئي Partial fraction بسهولة ويسر والتي تستلزم وقتاً كبيراً لعملها بالطرق التقليدية، هذا من الناحية الأكاديمية، أما من الناحية التطبيقية فيستطيع البرنامج العمل في جميع المجالات الهندسية مثل أنظمة التحكم Control Systems، وفي مجال الميكانيكا Mechanical Field، وكذلك محاكاة الإلكترونيات Electronics، وصناعة السيارات Automotive Industry، وكذلك مجال الطيران والدفاع الجوي Aerospace and Defense والكثير من التطبيقات الهندسية.

قامت شركة السيارات المرموقة نيسان Nissan بتخفيض وقت التطوير إلى ٥٠% عندما قامت باستبدال التصميم على الأوراق Paper Model Based Design إلى الأداة المتطورة في برنامج الماتلاب وهو Model Based Design.

فمع التقدم السريع في التكنولوجيا أصبحت الحاجة ملحة على تعلم مثل هذا البرنامج حتى نصبح في سباق التنافس الصناعي.

## مكونات شاشة الماتلاب

تنقسم واجهة البرنامج بالسهولة في التعامل معها، حيث يتم تقسيم مناطق العمل بها إلى ثلاث مناطق رئيسية، وهي كالتالي نافذة الأوامر Command Window، منطقة العمل Workspace وسجل الأوامر Command History، انظر الصورة التالية.



## نافذة الأوامر Command Window

تستخدم هذه الشاشة للتواصل مع برنامج الماتلاب من خلال وضع الاوامر بها. حيث يتم إدخال المدخلات Inputs والأوامر Commands، ويعمل الماتلاب على تحليل تلك البيانات ومدى مطابقة المدخلات للوظيفة المطلوبة منه، حتى تحصل على النتائج في نفس الشاشة.

## منطقة العمل Workspace

حيث يقوم الماتلاب بتسجيل المدخلات Inputs والمخرجات Outputs في هذه الشاشة. عند بدء العمل على الماتلاب لأول مرة، قد لا تظهر نافذة Workspace، وحتى تظهر اضغط بزر الفأرة على كلمة Workspace كما في الصورة التالية.

## نافذة تسجيل الأوامر Command History

يتم تسجيل كل ما يقوم به المستخدم على برنامج الماتلاب في هذه النافذة.

## قائمة ملف File

تتكون هذه القائمة من العديد من الخيارات، والتي تنفذ كل منها وظيفة محددة في البرامج.





## قائمة التعديل Edit

نجد تلك القائمة أوامر (نسخ Copy، قص Cut، لصق Paste، بحث Find) وكذلك ثلاث أوامر هامة هي:

- Clear Command Window
- Clear Command History
- Clear Workspace

حيث تعمل تلك الأدوات على مسح جميع المدخلات والنتائج من البرنامج.

Undo	Ctrl+Z
Redo	Ctrl+Y
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Paste Special...	
Select All	
Delete	
Find...	
Find Files...	
Clear Command Window	
Clear Command History	
Clear Workspace	

مسح قائمة الأوامر

مسح مسجل المدخلات والمخرجات

مسح منطقة العمل

## قائمة Debug

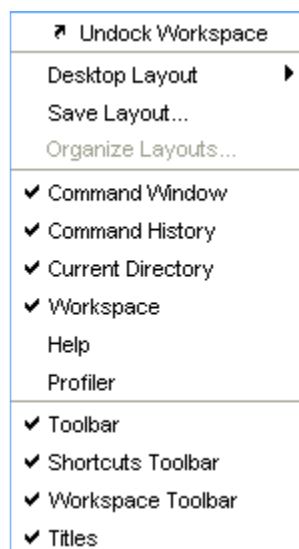
هذه القائمة خاصة بمعالجة البيانات، والطريقة المتبعة من قبل برنامج الماتلاب في مواجهة الأخطاء.

✓ Open M-Files when Debugging	
Step	F10
Step In	F11
Step Out	Shift+F11
Continue	F5
Clear Breakpoints in All Files	
Stop if Errors/Warnings...	
Exit Debug Mode	

تختص هذه المنطقة بعملية معالجة البيانات، وإحتمالات حدوث الخطأ في برنامج الماتلاب

## قائمة Desktop:

في هذه القائمة يتم التحكم بمحتوى الواجهة الخاصة ببرنامج الماتلاب، فمثلاً يمكننا إظهار نافذة الأوامر أو إخفائها، أنظر الصورة.



## قائمة Window

حيث يمكنك التنقل بين ملفات الماتلاب المختلفة، وكذلك النوافذ مثل نافذة الأوامر Command Window وغيرها الكثير.

Close All Documents		
0	Command Window	Ctrl+0
1	Command History	Ctrl+1
2	Current Directory	Ctrl+2
3	Workspace	Ctrl+3

## قائمة Help

حيث تقوم تلك القائمة، بتوفير المساعدات الضرورية في البرنامج، ووسائل الاتصال بالشركة المصنعة، وآخر التحديثات، وكذلك تعلم الماتلاب باللغة الإنجليزية.

Full Product Family Help	
MATLAB Help	F1
Using the Desktop	
Using the Command Window	
Web Resources	
Check for Updates	
Demos	
About MATLAB	

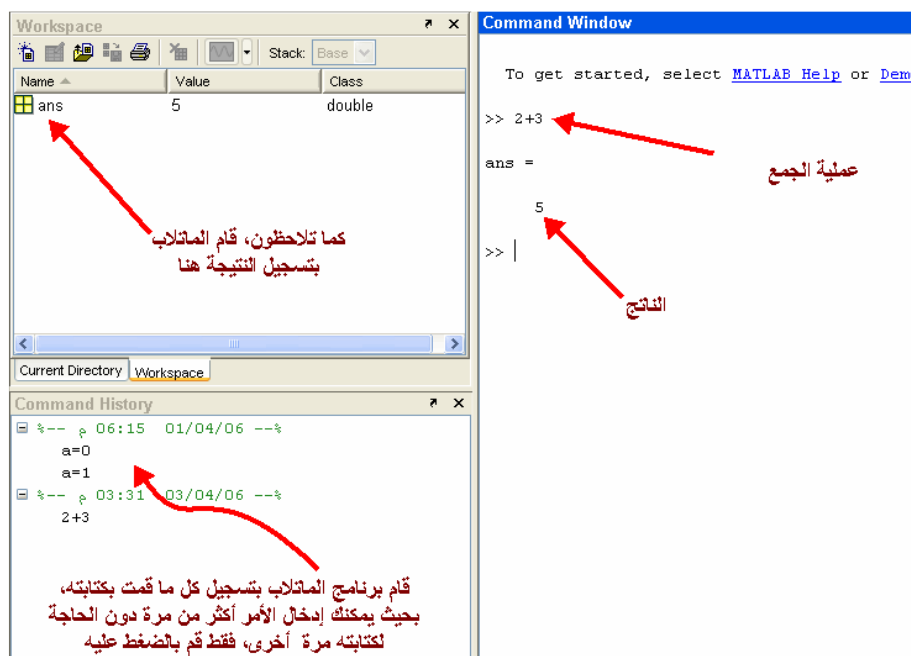
## العمليات الحسابية

هذا الجدول الذي يبين تعبير العملية الحسابية بواسطة الماتلاب وأولوياتها

الرمز	العملية	الصيغة في ماتلاب
^	ويعني القوة او الاس	A^B
*	الضرب	A*B

/	القسمة	A/B
+	الجمع	A+B
-	الطرح	A-B

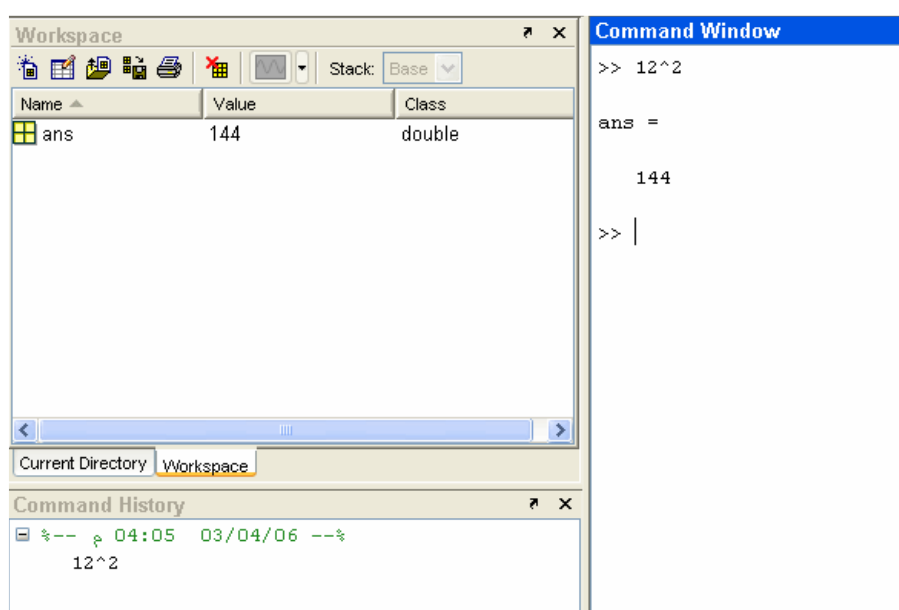
فمثلاً إذا قمنا بجمع 2+3 سيقوم الماتلاب بوضع الإجابة في صورة أرقام وهو 5، أنظر الصورة التالية.



اذهب إلى نافذة Workspace وقم بالنقر بالماوس بقرة مزدوجة، ستلاحظ ظهور نافذة حلت محل نافذة الأوامر وأصبحت نافذة الأوامر في الأسفل، أنظر الصورة.

## عملية وضع الأس

يأخذ رمز الأس ( ^ )، يمكن الحصول على هذا الرمز من خلال الضغط على Shift + 6 في لوحة المفاتيح، فمثلاً  $2^{12}=144$ ، أنظر الصورة.



## أخذ الجذر التربيعي

يتم أخذ الجذر التربيعي لأي رقم عن طريق كتابة الأمر sqrt، أنظر الصورة التالية.

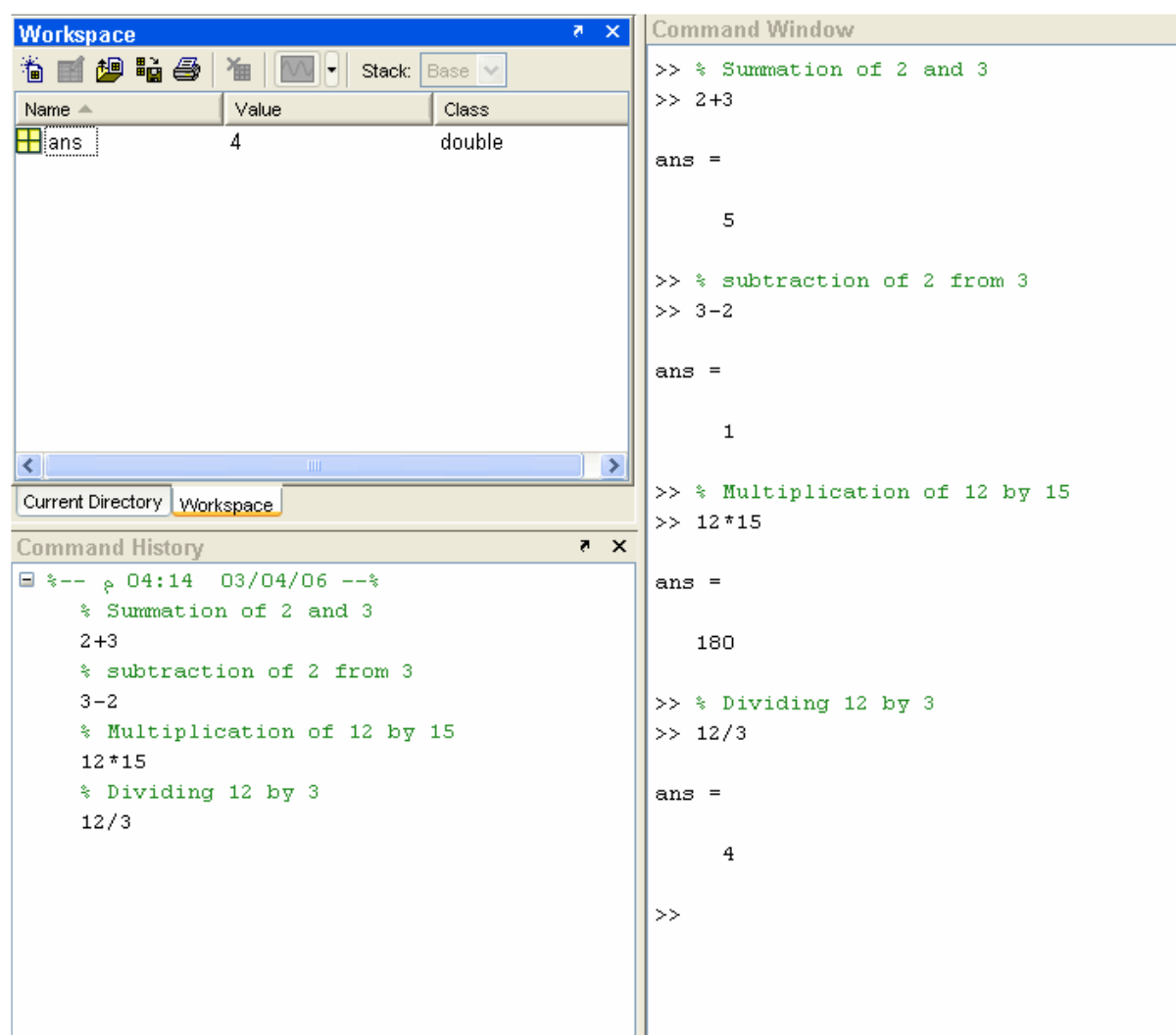
```
>> sqrt(144)
```

```
ans =
```

```
12
```

## وضع تعليقات أثناء البرمجة

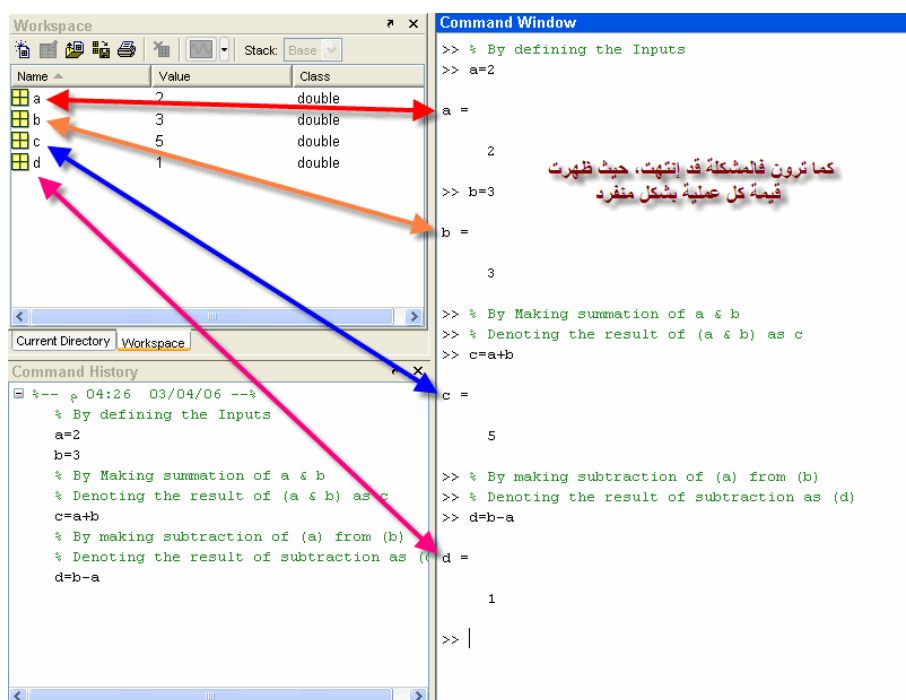
أما في برامج Qbasic و C++ وغيرها الكثير من برامج البرمجة، يتم وضع تعليقات لما نقوم به حيث تكون مثل المرجع لنا في معرفة ما نقوم به في جزء ما من البرنامج. في برنامج الماتلاب لوضع تعليق ما، لابد من أن نبدأ بوضع علامة مئوية ( % ) ثم نكتب ما نريده بعدها، لاحظ الصورة التالية.



## تعريف المتغير

ولكن كما تلاحظون فهناك مشكلة في نافذة Workspace، حيث أنها سجلت آخر قيمة فقط، وذلك لأن كل النتائج الأربعة تأخذ رمز ans حيث اننا لم نجعل لها رمزاً. لذلك يتم تعريف النتائج بحروف، بحيث يأخذ الحرف القيمة التي يدخلها المستخدم له.

يعطيك الماتلاب امكانية تعريف متغير مثل X واعطاءه قيمة معينة يبقى حاملا لها طول البرنامج مالم يتم تغييرها بعد ذلك.



كما ترون فالمشكلة قد انتهت تماماً، حيث تأخذ كل قيمة حرف معين.

## ملاحظات على المتغيرات

- الحروف Capital لا تساوي الحروف Small
- يجب ان يبدأ المتغير بحرف
- يمكن ان يكون طول المتغير حتى ٣٢ رمزاً
- لا يجوز ان يكون المتغير اسماً محجوزاً for, while, find
- بعض المتغيرات المعرفة مسبقاً في برنامج الماتلاب والمعروفة:

اسم المتغير	القيمة أو المعنى
ans	متغير متعرف دائماً وقيمته قيمة اخر جواب لم يتم تعريفه
pi	=3.1416π
inf	∞=Infinity ويعني المالا نهاية
Nan	عندما يكون جواب عملية حسابية معينة Nan تعرف انه الجواب غير معرف.
i, j	هو جذر -1- ويستخدم في الأرقام المركبة The complex variable v-1

#### Command Window

```
>> % the following process will show the infinity
>> 1/0
Warning: Divide by zero.

ans =

    Inf

>> % the following command will show Not A Number
>> 0/0
Warning: Divide by zero.

ans =

    NaN

>> % the following command will show the complex number
>> i

ans =

    0 + 1.0000i

>> % the following command will show the complex number
>> j

ans =

    0 + 1.0000i
```

#### Command Window

```
>> % The Following Command will show up the value of (pi)
>> pi

ans =

    3.1416

>> % The following command will show up the vlaue of (2*pi)
>> 2*pi

ans =

    6.2832

>> % the following Command will show up the value of square root of pi
>> sqrt(pi)

ans =

    1.7725
```

## الفاصلة المنقوطة نهاية الجملة

نلاحظ أننا كلما أدخلنا قيمة، أو حصلنا على نتيجة يقوم الماتلاب بإظهار القيمة المدخلة وكذلك النتيجة في نافذة الأوامر، مما يؤدي إلى كبر البرنامج المكتوب في حين أنه يؤدي مهمة بسيطة.

The screenshot shows the MATLAB Command Window with the following commands and outputs:

```
>> a=2
a =
    2
>> b=3
b =
    3
>> % By Making summation of a & b
>> % Denoting the result of (a & b) as c
>> c=a+b
c =
    5
```

Annotations in Arabic:

- Red arrow pointing to `a=2`: القيمة المدخلة (Entered value)
- Red arrow pointing to the output `a = 2`: يقوم الماتلاب بإظهار القيمة المدخلة (MATLAB displays the entered value)
- Red arrow pointing to the output `b = 3`: يقوم الماتلاب بإظهار القيمة المدخلة (MATLAB displays the entered value)
- Red arrow pointing to the output `c = 5`: يقوم الماتلاب أيضاً بإظهار النتائج بشكل مباشر (MATLAB also displays the results directly)

Text in the middle of the screenshot:

المشكلة أننا كلما أدخلنا قيمة ما، يقوم الماتلاب بإظهار القيمة المدخلة أو حتى النتيجة وهذا بالتالي يأخذ من مساحة الكتابة كما يبين أن البرنامج كبير جداً،

يمكن إخفاء القيمة المدخلة وكذلك النتيجة من الظهور (ولكن عملية إدخال النتيجة والجمع مثلاً تتم بشكل طبيعي ويقوم الماتلاب بتنفيذ ما يأمره المستخدم) عن طريق وضع علامة ( ; ) بعد كل قيمة مدخلة أو بعد طلب نتيجة ما (الجمع مثلاً) ويتم إظهار النتيجة أو القيم المدخلة إذا طلب المستخدم ذلك، عن طريق وضع حرف المدخلات أو النتيجة المطلوبة دون استخدام الرمز المذكور ( ; ).

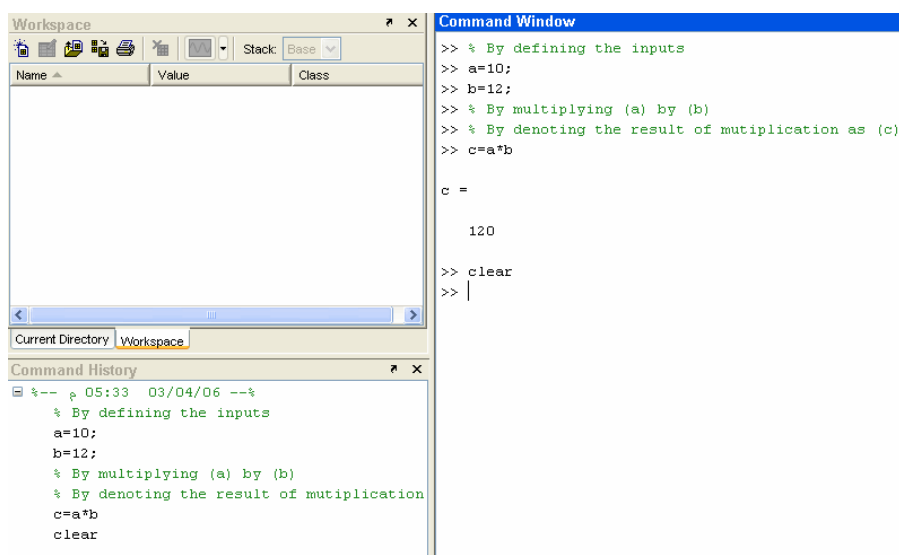
```
>> x=9
x =
    9
>> x=9;
>>
```

Annotations in Arabic:

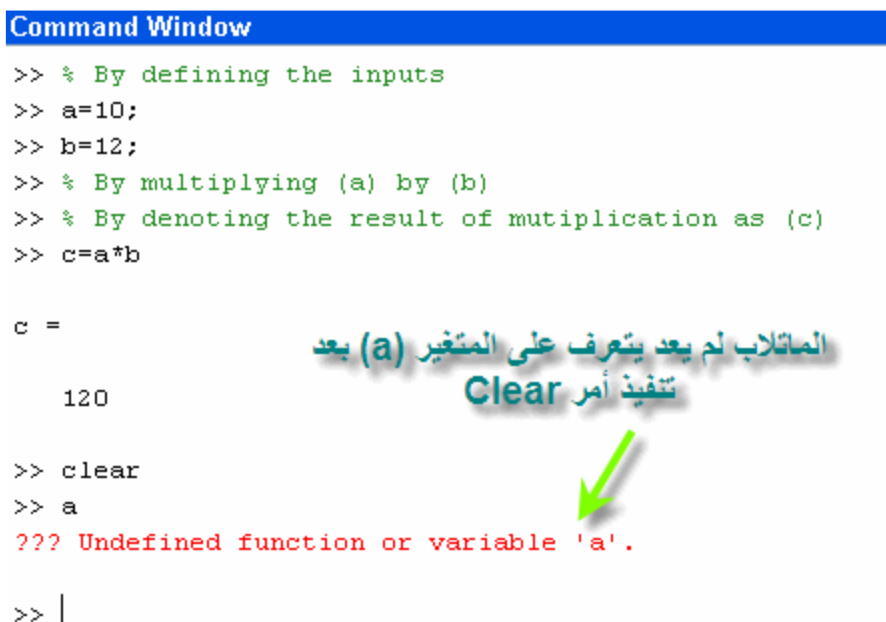
- Red arrow pointing to the output `x = 9`: اظهر النتيجة (Show the result)
- Red arrow pointing to the output `x = 9`: عدم اظهارها (Do not display it)

## إلغاء القيم المدخلة والنتائج

يمكن للماتلاب مسح القيم المدخلة والنتائج (والتي تسجل في نافذة تسجيل النتائج)، دون مسح ما قمت بكتابته، وذلك باستخدام أمر `Clear`، فبعد تنفيذ أمر، أنظر الصورة التالية.



وللتأكد قم بوضع أي حرف من الحروف التي قمت بتعريفها مسبقاً للماتلاب، ستلاحظ ان الماتلاب لا يتعرف عليها الآن، أنظر الصورة.



## عملية المسح الجزئي للمتغيرات

ليس شرطاً أن نقوم بعملية مسح كلي لكل البرنامج بل من الممكن عمل مسح لمتغير واحد فقط، عن طريق كتابة أمر Clear ثم اسم المتغير.



**Workspace**

Name	Value	Class
b	12	double
c	120	double

لا توجد قيمة للمتغير (a)، بعد تنفيذ أمر (Clear a)

**Command Window**

```
>> % By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of mutiplication as (c)
c=a*b

c =
    120

>> % By clearing the value of a
>> clear a
>> a
??? Undefined function or variable 'a'.

>> b
b =
    12

>>
```

فمنما يمسح قيمة (a) فقط ، لاحظ اختفاء قيمة (a) من نافذة Workspace

لا تزال قيمة المتغير (b) موجودة ، بينما لا توجد قيمة للمتغير (a)

**Command History**

```
%-- م 05:43 03/04/06 --%
% By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of mutiplication
c=a*b
% By clearing the value of a
clear a
a
b
```

## الأوامر الخاصة ببرنامج الماتلاب

نستكمل بعض الأوامر الخاصة ببرنامج الماتلاب

## الدوال المثلثية Trigonometric functions

Built In Function	Trigonometric Function
sin	Sine
cos	Cosine
tan	Tangent

ملاحظة: يقوم الماتلاب بقياس الزوايا بالتقدير الدائري Radian

The screenshot shows the MATLAB environment. The **Workspace** window on the left lists variables `v`, `x`, and `y`, all of type `double` with a value of `1`. The **Command Window** on the right shows the following commands and outputs:

```
>> % Defining the Sine function
>> x=sin(pi/2)

x =

    1

>> % Defining the Cosine Function
>> y=cos(2*pi)

y =

    1

>> % Defining the Tangent Function
>> v=tan(pi/4)

v =

    1.0000

>> |
```

Arrows point from the text "يتم التعويض بقيمة مختلفة للزاوية في الدوال المثلثية" (The trigonometric functions are substituted with different values for the angle) to the arguments `pi/2`, `2*pi`, and `pi/4` in the commands.

## الدوال المثلثية العكسية Inverse Trigonometric functions

Built In Function	Trigonometric Function
asin	Inverse Sine
acos	Inverse Cosine
atan	Inverse Tangent

```
>> % By defining the Inverse sine function
>> a=asin(1)
```

**يمكننا تعريف الدوال المثلثية العكسية بالطريقة التالية، ماهي قيمة الزاوية التي إذا أخذنا لها Sine نحصل على العدد 1 بالتأكيد ستكون  $(\pi/2)=1.5708$**

**a =**  
**1.5708**

```
>> % By defining the Inverse Cosine Function
```

```
>> b=acos(1)
```

**نحصل على زاوية مقدارها صفر أو  $\pi/2$ ، إذا أخذنا Inverse Cosine للعدد 1**

**b =**  
**0**

```
>> % By defining the Inverse Tangent function
```

```
>> c=atan(1)
```

**الزاوية المناظرة لدالة المماسية العكسية للعدد واحد هي  $\pi/4=0.7854$**

**c =**  
**0.7854**

## الدالة الأسية Exponential Function

تكتب الدالة الأسية في ماتلاب في صيغة الأمر `exp(x)`

$$\exp(x) = e^x$$

```
>> % By applying the exponential function for a parameter x
>> % By defining the parameter y
>> syms y
>> x=1

x =

    1

>> y=exp(x)

y =

    2.7183
```

### الجذر التربيعي sqrt

تكتب دالة الجذر التربيعي في ماتلاب في صيغة الأمر  $\text{sqrt}(x)$

$$\text{sqrt}(x) = \sqrt{x}$$

### الأعداد المركبة Complex Numbers

تأخذ الأعداد المركبة صيغة واحدة وهي تواجد جزء العدد الحقيقي Real number وجزء العدد التخيلي Imaginary Numbers، وتكون في الصيغة التالية

$$z = x + y * i$$

### الكتابة فوق قيمة العدد المركب

يمكننا الكتابة فوق قيمة العدد (i)، أي تغيير قيمته، حيث سنقوم بوضع قيمة لهذا الرمز.

```
Command Window
>> % Overwriting the complex variable i
>> i=3;
>> a=1+3*i

a =

    10

>> % Notice that the presence of (*) has dealt (i) not complex but the value
>> % by the user
>> % If the multiplication sign has been removed so (i) represents complex No.
>> b=1+3i

b =

    1.0000 + 3.0000i
```

يتم في برنامج الماتلاب العديد من العمليات على الأعداد المركبة مثل:

- اختيار العدد الحقيقي فقط، ونستخدم لذلك الأمر  $\text{real}(z)$
- اختيار العدد التخيلي فقط، ونستخدم لذلك الأمر  $\text{imag}(z)$

- إيجاد الزاوية Phase Angle، ونستخدم لذلك الأمر  $\text{angle}(z)$  وهو مناظر للعلاقة التالية

$$\text{angle} = \tan^{-1}\left(\frac{\text{Imaginary Number}}{\text{Real Number}}\right)$$

- إيجاد القيمة المطلقة، ونستخدم لذلك الأمر  $\text{abs}(z)$  وهو مناظر للعلاقة التالية

$$\text{Absolute Value} = \sqrt{X^2 + Y^2}$$

- جمع عددين مركبين، ويتم ذلك عن طريق جمع الأعداد الحقيقية مع بعضها، وجمع الأعداد المركبة مع بعضها.

```
>> % Writting a complex number and performing its operations
```

```
>> z=3+4i
```

العدد الحقيقي      العدد التخيلي

z = 3.0000 + 4.0000i

```
>> % By selecting the Real Part using (real) command
```

```
>> real(z)
```

ans = 3

يستخدم الأمر Real يتم إختيار العدد الحقيقي فقط من العدد المركب حيث يكون 3 في المثال الموضح

```
>> % By Selecting the Imaginary Part using (imag)command
```

```
>> imag(z)
```

ans = 4

يتم إختيار العدد التخيلي فقط من خلال إستخدام الأمر Imag حيث يكون 4 في هذا المثال

```
>> % By Getting the phase Angle using the (angle) command
```

```
>> angle(z)
```

ans = 0.9273

الزاوية الطور

```
>> % Getting the absolute value of complex number using (abs) command
>> abs(z)

ans =

    5

>> % By defining another complex number called v
>> v=2+3i

v =

    2.0000 + 3.0000i

>> z+v

ans =

    5.0000 + 7.0000i

>> % By getting the phase angle using the (atan2) command
>> angle=atan2(imag(z),real(z))

angle =

    0.9273
```

استخدام القيمة المطلقة

جمع عددين مركبين

أمر الجزء الحقيقي للعدد المركب Z

أمر الجزء التخيلي للعدد المركب Z

حصلنا على نفس الزاوية السابقة أيضاً

## اللوغاريتمات الطبيعية Natural Logarithm

يرمز الماتلاب للوغاريتمات الطبيعية بالرمز  $\log(x)$

## العمليات التقريبية لأعداد واقعة بين رقمين

أي رقم عشري يمتاز بأنه واقع بين رقمين صحيحين، فالماتلاب له القدرة على اختيار أحد هذين الرقمين باستخدام الأمرين Ceil لاختيار الرقم الأكبر، والأمر Floor لاختيار الرقم الأصغر.

```
>> % Selection the integer numbers limiting a fractional number.
>> a=5.6

a =

    5.6000

>> ceil(a)

ans =

    6

>> floor(a)

ans =

    5
```

تحديد قيمة العدد العشري

اختيار العدد الصحيح الأكبر من خلال الأمر Ceil

اختيار العدد الصحيح الأصغر من خلال الأمر Floor

مثال ١:

أكتب ما يلي باستخدام ماتلاب:

$$١. \quad 6 \frac{10}{13} + \frac{18}{5(7)} + 5(9)^2$$

$$٢. \quad 6 \left( 35^{\frac{1}{4}} \right) + 14^{0.35}$$

الحل:

$$١. \quad 6*(10/13)+18/(5*7)+5*9^2$$

$$\text{ans}=410.129$$

$$٢. \quad 6*35^{(1/4)}+14^{0.35}$$

$$\text{ans}=17.1123$$

مثال ٢:

أوجد قيمة  $z$  إذا كنت تعرف أن  $x=2$ ،  $y=17$

$$z = x^3 + y^2 + \frac{3\pi}{x^2 + y^2}$$

الحل:

بداية نعرف القيم المتغيرة و ثم نكتب المعادلة على الماتلاب

```
x=2
y=17
z=x^3+y^2+(3*pi/(x^2+y^2))
z=17.2
```

مثال ٣:

أوجد القدرة  $P$  إذا كنت تعرف أن التيار  $I=10A$  والمقاومة  $R=10\Omega$ ، علماً بأن  $P=IV$ ،  $V=IR$ .

الحل:

```
R=10;
I=10;
V=I*R;
P=I*V
P=1000
```

مثال ٤:

حجم اسطوانة دائرية بطول  $h$  ونصف قطر  $r$  يعطى بالعلاقة التالية

$$v = \pi r^2 h$$

عندنا أسطوانة دائرية طولها 15 متر ونصف قطرها 8 متر. نريد إنشاء أسطوانة أخرى بحجم أكبر من الأولى بنسبة 20% ولها نفس طول الأولى، كم يجب أن يكون نصف القطر.

الحل:

أولاً نعرف المتغيرات وثم نكتب المعادلات

```
R=8;
H=15;
V=pi*R^2*H;
V=V+0.2*V
R=(V/pi/H)^0.5
R=78.8
```

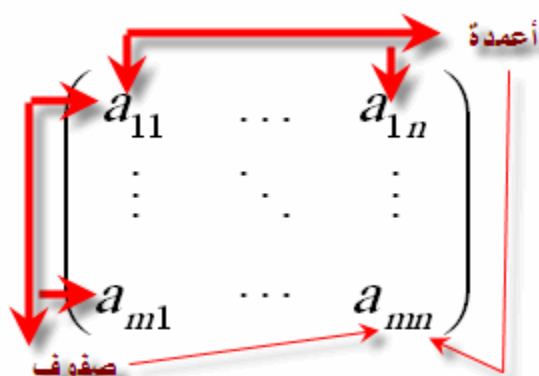
ملاحظه: استخدمنا الفاصلة المنقوطة بدل الفاصلة العادية حتى لا يتم عرض النتائج المرحلية.

## الفصل الثاني المصفوفات Matrices



## تعريف المصفوفة

هي مجموعة من البيانات والتي يتم وضعها في صورة صفوف وأعمدة، وتأخذ الشكل التالي



وتستخدم المصفوفات في حل كثيرات الحدود Polynomials، وفي حل مجموعة من المعادلات كما سيتم شرحه لاحقاً.

## كيفية كتابة المصفوفات في برنامج الماتلاب

يتم إدخال المصفوفة بكتابة عناصر الصف الأول، ثم الثاني وهكذا. فمثلاً كتابة مصفوفة مثل التالية

$$\begin{pmatrix} 1 & 3 \\ 6 & 4 \end{pmatrix}$$

ولكن قبل إدخال القيم التالية، على الجميع أن يعلم بأنه يتم كتابة عناصر الصف الأول، ويتم الفصل بين أرقام الصف الأول إما بفاصلة ( , ) أو بعمل مسافة بين الأرقام، بعد إدخال قيم الصف الأول يتم فصل عناصر الصف الأول عن عناصر الصف الثاني (الذي سيتم إدخال قيمه) إما بالضغط على مفتاح Enter أو باستخدام الفاصلة المنقوطة ( ; ). لاحظ ان كل مصفوفه تبدأ بقوس مربع وتنته به.

```
>> x=[4 5 6]
x =
     4     5     6
```

مصفوفة من صف واحد تم تعريفها بدون فواصل

```
>> x=[3,3,5]
x =
     3     3     5
```

التعريف باستخدام الفواصل

```
>> x=[4 5 ; 3 5; 67 4]
x =
     4     5
     3     5
    67     4
```

التفريق بين الأعمدة باستخدام فاصله منقوطة

```
>> x=[45 6
35 64
35 67]
x =
    45     6
    35    64
    35    67
```

التفريق بين الأعمدة بواسطة  
enter  
كل صف يكتب يتم بعده الضغط على  
enter

## العمليات الأساسية التي تتم على المصفوفات

- الجمع
- الطرح
- الضرب
- القسمة
- المصفوفة الأسية

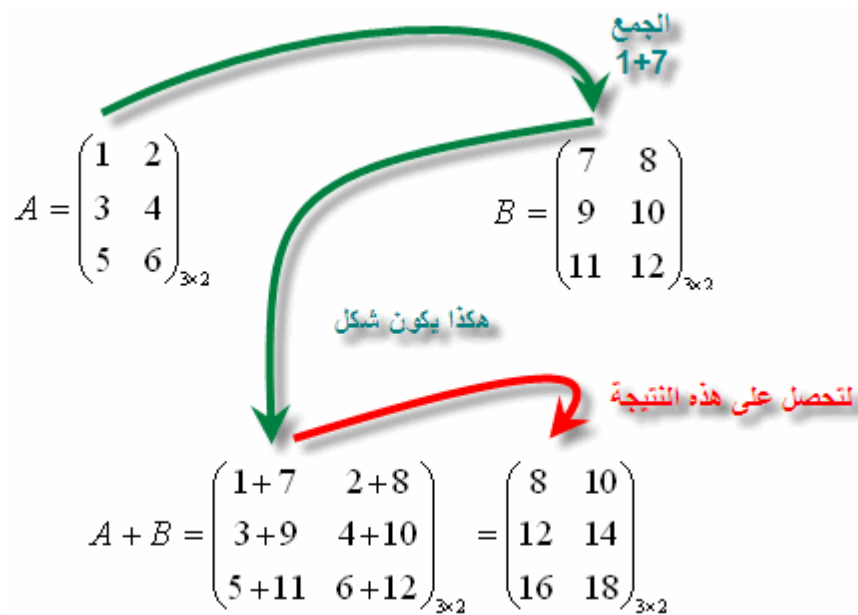
## الجمع والطرح

يجب أولاً أن نذكر شرط جمع أو طرح مصفوفتين. لنفترض أن لدينا مصفوفتين، A و B فشرط جمعهما أو طرحهما أن يكون كلاهما له نفس عدد الصفوف m، وكذلك نفس عدد الأعمدة n. فمثلاً المصفوفتان التاليتان يمكن جمعهما أو طرحهما لأنها يحملان نفس عدد الصفوف والأعمدة.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \quad \begin{array}{l} \text{كما ترى فإن عدد} \\ \text{الصفوف في} \\ \text{المصفوفة الأولى} \end{array}$$

$$B = \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix} \quad \begin{array}{l} \text{مساوياً لعدد الصفوف} \\ \text{في المصفوفة الثانية،} \\ \text{وكذلك عدد الأعمدة} \\ \text{لكلتا المصفوفتين} \end{array}$$

كيف تتم عملية جمع أو طرح مصفوفتين:



في ماتلاب يجب أولاً كتابة المصفوفتين A و B ثم استخدام رمز الجمع ( + ) لتتم عملية الجمع.

```
>> % Today We're going to discuss the basic operation on Matrices
>> % By Defining the Matrix A
>> A=[1 2;3 4;5 6]
```

A =

```
1    2
3    4
5    6
```

```
>> % By Defining the matrix B
>> B=[7 8;9 10;11 12]
```

B =

```
7    8
9    10
11   12
```

```
>> % By making addition to both A&B
```

```
>> % Assume that the Result of summation would be denoted as C
>> C=A+B
```

C =

```
8    10
12   14
16   18
```

مثال:

دائرة كهربائية فيها خمس مقاومات على التوالي 230، 543، 623، 75، 90. تم زيادة جميع المقاومات بمقادير 170.

الحل:

تعريف مصفوفه أولى بقيمة المقاومات ثم جمع المصفوفة مع 170.

```
>> x=[230,543,623,75,90 ]

x =

    230    543    623    75    90

>> x+170

ans =

    400    713    793    245    260
```

### ضرب المصفوفات

شرط ضرب أي مصفوفتين هو أن يكون عدد أعمدة المصفوفة الأولى  $n1$  مساوياً لعدد الصفوف في المصفوفة الثانية  $m2$ .

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 8 \end{pmatrix}_{3 \times 2}$$

$$B = \begin{pmatrix} 0 & 3 & 3 \\ 4 & 9 & 7 \end{pmatrix}_{2 \times 3}$$

$$C = A \times B = \begin{pmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 8 \end{pmatrix}_{3 \times 2} \times \begin{pmatrix} 0 & 3 & 3 \\ 4 & 9 & 7 \end{pmatrix}_{2 \times 3}$$

$$C = \begin{pmatrix} (1 \times 0) + (2 \times 4) & (1 \times 3) + (2 \times 9) & (1 \times 3) + (2 \times 7) \\ (4 \times 0) + (6 \times 4) & (4 \times 3) + (6 \times 9) & (4 \times 3) + (6 \times 7) \\ (9 \times 0) + (8 \times 4) & (9 \times 3) + (8 \times 9) & (9 \times 3) + (8 \times 7) \end{pmatrix}_{3 \times 3}$$

$$C = \begin{pmatrix} 8 & 21 & 17 \\ 24 & 66 & 54 \\ 32 & 99 & 83 \end{pmatrix}_{3 \times 3}$$



لنقوم الآن بإدخال نفس المثال على الماتلاب

```

Command Window
>> % By defining the Matrix A
>> A=[1 2;4 6;9 8];
>> % By Defining the Matrix B
>> B=[0 3 3;4 9 7];
>> % C=A*B
>> C=A*B

C =

    8    21    17
   24    66    54
   32    99    83
>>

```

إذا كان عدد أعمدة المصفوفة الأولى لا يساوي عدد صفوف المصفوفة الثانية

```
>> x=[3 45 4; 5 6 2; 2 34 5]
```

```
x =
```

```

    3    45     4
    5     6     2
    2   34     5

```

```
>> y=[ 2 3 4 ; 5 3 2 ]
```

خطأ..  
لماذا؟؟؟

```
y =
```

```

    2     3     4
    5     3     2

```

```
>> x*y
```

```

??? Error using ==> mtimes
Inner matrix dimensions must agree.

```

## ضرب المصفوفة عنصر لعنصر

يمكن باستخدام الماتلاب ضرب عنصر لعنصر بالمصفوفة أو قسمة عنصر لعنصر أيضا وذلك بوضع نقطه قبل اشارة الضرب أو القسمة.

```
>> x=[ 3 4 6 ];
>> y=[2 5 6];
>> x*y
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> x.*y

ans =

        6        20        36

>> x./y

ans =

    1.5000    0.8000    1.0000

>> x.^y

ans =

         9       1024      46656
```

### قسمة المصفوفات

قد يستغرب البعض من وجود كلمة القسمة للمصفوفات، ولكن الحقيقة أنها موجودة ومستخدمة بكثيرة ولكننا لا ننتبه لوجودها، فبهذه القسمة نقوم بحل المعادلات كثيرة الحدود. لنفترض أن لدينا معادلتان كالتالي.

$$3X + 3Y = 3$$

$$2X + 3Y = 5$$

وكلتا المعادلتان يمكن حلها ليكون الناتج  $X=-2$  و  $Y=3$

يمكن وضع المعادلتان في صورة مصفوفة كما في الشكل التالي.

$$\begin{pmatrix} 3 & 3 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

المعادلتان في صورة  
المصفوفة

نجد أنه يمكننا أن نضعها في الصيغة التالية.

$$AX = B$$

وبالتالي من أجل الحصول على X يجب قسمة A على B كما في الصورة التالية.

$$X = \frac{B}{A}$$

ولكن ماذا تعني  $\frac{1}{A}$  من ناحية المصفوفات وليست الأعداد؟

$$\frac{1}{A} = inv(A)$$

وهذا ما يسمى قسمة المصفوفات

ولكن يشترط عند إيجاد  $inv$  أن تكون المصفوفة مربعة (أي عدد الصفوف يساوي عدد الأعمدة)، وبالتالي يمكن إيجاد قيمة  $X$  و  $Y$  عن طريق وضع المعادلة في الصورة التالية، مع الأخذ في الاعتبار أن تتوفر شرط عملية الضرب بين المصفوفتين.

$$\begin{pmatrix} X \\ Y \end{pmatrix} = inv \left( \begin{pmatrix} 3 & 3 \\ 2 & 3 \end{pmatrix} \right) \times \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

2x2      2x1

يجب الإنتباه لشرط عملية ضرب المصفوفة

فإذا قمنا بكتابة المعادلتين في الماتلاب كما في الصورة السابقة

```
Command Window
>> % By defining the Coefficient Terms
>> A=[3 3;2 3];
>> % By Defining the Absolute Terms
>> B=[3;5];
>> C=inv(A)*B

C =
-2
 3
>>
```

كما ترى فلقد حصلنا على نفس القيم التي حصلنا عليها باستخدام طريقة الحذف

**X=-2**

**Y=3**

مثال:

اوجد قيم  $x$ ،  $y$  و  $z$  التي تحقق المعادلات التالية:

$$4x+12y+23z=12$$

$$34x+4y-3z=-90$$

$$x-32y-9z=0$$

هذه المعادلات يُكن التعبير عنها بواسطة مصفوفه كالتالي

$$\begin{bmatrix} 4 & 12 & 23 \\ 34 & 4 & -3 \\ 1 & -32 & -9 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 12 \\ 90 \\ 0 \end{bmatrix}$$

اذن قيمة قيم x، y و z تساوي المصفوفة التي بعد اشارة اليساوي تقسيم المصفوفة المضروبة بالمتغيرات ويمكن حل ذلك بالماتلاب كما يلي:

```
>> s=[4 12 23 ; 34 4 -3; 1 -32 -9];           مصفوفة المتغيرات
>> w=[12; 90; 0];                             العمود على يمين المساواة
>> inv(s)*w

ans =

    2.6401
    0.0761
    0.0229                                     الجواب

>> w/s
??? Error using ==> mrdivide
Matrix dimensions must agree.                لا يمكن تقسم مصفوفة على مصفوفة
```

### الفرق بين المتجهات والمصفوفات

المتجهات هي مصفوفة ولكن إما بعمود واحد Column Vector أو صف واحد Row Vector. فمثلاً الصورة التالية لمتجه صفي.

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

```
A =
```

```
1 2 3 4 5 6 7 8 9 10
```

Row Vector

وهذه صورة لمتجه عمودي.



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

B =

```
1
2
3
4
5
6
7
8
9
10
```

Column Vector

أما المصفوفة فهي التي يزيد عدد صفوفها وأعمدتها عن صف واحد أو عمود واحد.

### العمليات الشائعة على المتجهات

١. طول المتجه
٢. إضافة عنصر
٣. استبدال عنصر
٤. عملية حذف عنصر
٥. نداء عنصر
٦. نداء عدد عناصر
٧. إيجاد العنصر الأكبر
٨. إيجاد العنصر الأصغر
٩. إيجاد حاصل ضرب العناصر

### طول المتجه

لنقوم بتعريف متجه صفي لدى الماتلاب كما في الصورة التالية

Command Window

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

A =

```
1 2 3 4 5 6 7 8 9 10
```

تعريف متجه صفي

والآن نقوم بالعملية الأولى وهي طول المتجه

Command Window

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

```
>> % It's required to get the length of A
```

```
>> length(A)
```

ans =

```
10
```

فالمقصود بـ **length** هو عدد العناصر الموجودة في المتجه وكما هو واضح أن عدد العناصر هو ١٠

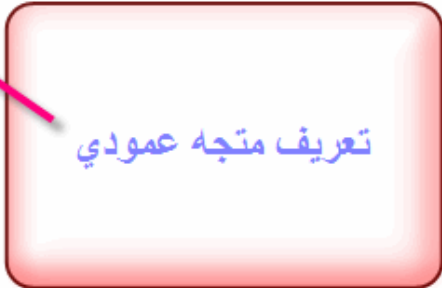
## إضافة عنصر لمتجه

لنقوم بوضع متجه عمودي في الماتلاب كما يلي

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

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
```




لنقل إننا نريد إضافة الرقم 120 في الخانة الحادية عشرة، أي الخانة التالية للخانة العاشرة.

```
A =


     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(11)=120
```



```
A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
    120
```



ملاحظة: في المثال السابق تمت إضافة الرقم 120 إلى الخانة 11، فماذا إذا قمنا بإضافة رقم جديد ولكن في الخانة رقم 13، فماذا ستكون قيمة الخانة 12 التي لم يتم إضافة أي عنصر لها.

```

Command Window

1
2
3
4
5
6
7
8
9
10
120
>> A(13)=140

A =

1
2
3
4
5
6
7
8
9
10
120
0
140

```

تمت إضافة العنصر ١٤٠ إلى الخانة رقم ١٣

كما ترى فإن الماتلاب يفترض قيمة الخانة ١٢ بصفر، وعلى الرغم من عدم إدخالنا لقيمتها، لذلك نستنتج أن أي خانة تقوم بتخطيها يقوم الماتلاب بفرض قيمتها بصفر

### إضافة أكثر من عنصر متتالي لمتجه

لنفترض أننا نريد إضافة مجموعة من العناصر المتتالية في الخانات 11، 12، و 13 ويمكن بدلاً من إدخال كل رقم على حدى، إذا أردنا كتابة مجموعة من الأرقام المتتالية من 11 إلى 13 نكتب 11:13 وبالتالي تكون الكتابة في الماتلاب

```
Command Window
>> A=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10];
>> A(11:13)=[11 12 13]

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
    11
    12
    13
```

يتم تحديد قيم الخانات  
بشرط أن يتم وضعها في  
قوسين

-----

[قيم الخانات]

تم تحديد الخانات المتتالية  
من ١١ إلى ١٣

```
Command Window
>> A=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10];
>> A(11:13)=[11 12 13]

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
    11
    12
    13
```

يتم تحديد قيم الخانات  
بشرط أن يتم وضعها في  
قوسين

-----

[قيم الخانات]

تم تحديد الخانات المتتالية  
من ١١ إلى ١٣

### استبدال عنصر في متجه

إذا أردنا أن نستبدل العنصر الثالث بدلاً من الرقم 3 إلى الرقم 15 آل ما علينا فعله هو كتابة  $A(3)=15$

## استبدال مجموعة عناصر متتالية في متجه

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

A =
     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(6:10)=[0 0 0 0 0]

A =
     1
     2
     3
     4
     5
     0
     0
     0
     0
     0
```

مجموعة العناصر في المتجه

تم تحديد مجموعة العناصر التي سيتم تغييرها

مجموعة العناصر بعد

## حذف عنصر من المتجه

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

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(10)=[]

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
```

تم تحديد العنصر العاشر لحذفه

يتم وضع قوس مربع فارغ ليبدل على أن هذه عملية حذف للعنصر

كما ترى اختفاء العنصر العاشر

### حذف مجموعة عناصر متتالية في متجه

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

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(6:10)=[]

A =

     1
     2
     3
     4
     5
```

تم تحديد مجموعة العناصر المطلوب حذفها

كما تلاحظ اختفاء مجموعة العناصر التي تم تحديدها

### نداء عنصر من متجه

نداء عنصر المقصود به هو الحصول على قيمة العنصر في أي مكان من المتجه

```

Command Window

To get started, select MATLAB Help or Demos from the Help menu.

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

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(5)

ans =

     5
    
```

نداء العنصر رقم ٥ وقيمته  
٥ كما هو واضح



### نداء أكثر من عنصر من متجه

للحصول على قيم مجموعة عناصر محددة من متجه، قم بعمل الآتي على نافذة الأوامر.



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

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(6:10)

ans =

     6
     7
     8
     9
    10
```

تم تحديد مجموعة  
العناصر الذين نريد  
الحصول على قيمهم  
داخل المتجه

**إيجاد العنصر الأكبر في المتجه باستخدام الأمر max**  
لإيجاد العنصر الأكبر في متجه، يتم استخدام الأمر max.

```
Command Window
>> A=[10 22 36 41 44 59 61 73];
>> max(A)

ans =

    73

>>
```

١ - يجب عند إيجاد الرقم الأكبر  
داخل المتجه كتابة الأمر max  
ويجب أن يأخذ الصورة التالية  
( اسم المتجه ) max

٢ - وهذا هو الرقم الأكبر داخل  
المتجه

### إيجاد العنصر الأصغر في المتجه باستخدام الأمر min

```
Command Window
>> A=[10 22 36 41 44 59 61 73];
>> min(A)

ans =

    10
```

١ - لإيجاد العنصر الأصغر في المتجه،  
قم باستخدام الأمر min  
حيث يأخذ الصورة التالية  
min(اسم المتجه)

٢ - كما ترى فإن العنصر  
الأصغر في هذا المتجه هو

### إيجاد مجموع عناصر المتجه باستخدام الأمر sum

```
Command Window
>> Y=[1 2 3];
>> sum(Y)

ans =

     6
```

### إيجاد حاصل ضرب العناصر في المتجه باستخدام الأمر prod

```
Command Window
>> Y=[1 2 3 4];
>> prod(Y)

ans =

    24
```

### إنشاء متجه باستخدام الأمر linspace

هذا الأمر يتيح لك إنشاء مصفوفة بصف واحد تبدأ برقم وتنته برقم وتحديد عدد عناصرها. ويمكن تعريف هذا الأمر أيضاً بأنه يستخدم في عملية إنتاج متجه، عن طريق تحديد الرقم الأصغر والرقم الأكبر، وعدد النقط المرغوبة بين هذين الرقمين، ويأخذ الصورة التالية.

`linspace(minimum number, maximum number, number of point in between)`

فالمثال التالي يوضح كيفية استخدام 10 نقاط من بين الرقمين 5 و15.

```

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m
1 - clc
2 - clear
3 - A=linspace(5,15,10)
4

Command Window

A =

Columns 1 through 9

5.0000 6.1111 7.2222 8.3333 9.4444 10.5556 11.6667 12.7778 13.8889

Column 10

15.0000

>>
    
```

وهذا ما يسمى النظام العشوائي للماتلاب Randomization System in Matlab، والميزة في هذا النظام هو أن المتجه نظام عشوائي متزايد.

### إنشاء متجه باستخدام الأمر randint

هذا الأمر من ضمن الأوامر والتي تنشأ نظام عشوائي للأرقام، ولكن ليس على هيئة كسور مثل الأمر السابق، كما أن نظام الأرقام به ليس تزايدياً أو تناقصياً بل عشوائياً ويأخذ الصورة التالية

randint(number of rows, number of columns, [minimum number, maximum number])

وهذا مثال بسيط باستخدام هذا الأمر على الماتلاب

```

1 - clc
2 - clear
3 - A=randint(1,10,[5 15])
4 - |

```

Command Window

A =

15 7 11 10 14 13 10 5 14 9

>>

كما ترى فهذه أرقام صحيحة، ولكن عشوائية الاختيار من قبل الماتلاب

## العمليات الشائعة على المصفوفات

### إيجاد حجم المصفوفة باستخدام الأمر size

لإيجاد حجم المصفوفة أو دعونا نقول لإيجاد عدد الصفوف والأعمدة لمصفوفة، يجب استخدام الأمر size، حيث لا يصلح استخدام الأمر length. ولتوضيح الأمر دعونا نقوم بعمل مثال مبسط لشرح هذا الأمر، أولاً لنقوم بعمل مصفوفة غير منتظمة (أي أن عدد الصفوف لا يساوي عدد الأعمدة).

```
>> A=[3 4 9;2 4 5]
```

A =

```

3  4  9
2  4  5

```

والآن لنقوم بكتابة الأمر size لمعرفة حجم المصفوفة.

```
>> A=[3 4 9;2 4 5]
```

```
A =
```

```
3 4 9
2 4 5
```

الأمر size

```
>> size(A)
```

عدد الصفوف

```
ans =
```

```
2 3
```

عدد الأعمدة

أما إذا أردنا أن نعرف عدد الصفوف فقط نقوم بكتابة التالي.

```
>> size(A,1)
```

```
ans =
```

```
2
```

أما إذا أردنا أن نعرف عدد الأعمدة فقط نقوم بكتابة التالي

```
>> size(A,2)
```

```
ans =
```

```
3
```

### إضافة عنصر إلى المصفوفة

لنقوم أولاً بتعريف مصفوفة في الماتلاب

```
>> 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
```

لنفترض أننا نريد أن نقوم بوضع رقم 42 في الصف الثاني والعمود الخامس، نقوم بكتابة التالي في الماتلاب.

```
>> B(2,5)=42
```

```
B =
```

```
1  3  7  8  0
2  6  5 11 42
12 14 15 13  0
```

كما تلاحظ فإن الصف الأول والصف الثالث للعمود الخامس، لم يتم وضع قيم بهما، لذلك قام الماتلاب بافتراضهما صفراً.

إذا كنا نريد إضافة الأعداد 31 و 54 و 13 و 11 في الصف الرابع والعمود الأول والثاني والثالث والرابع على التوالي، يمكن ذلك من خلال الماتلاب بالشكل التالي

```
>> B(4,1:4)=[31 54 13 11]
```

الأعمدة من الأول إلى الرابع

```
B =
```

الصف الرابع

```
1  3  7  8
2  6  5 11
12 14 15 13
31 54 13 11
```

العناصر الجديدة

### استبدال عنصر في مصفوفة

لاستبدال عنصر أو عدة عناصر داخل المصفوفة، نقوم أولاً بتعريف مصفوفة كما ذكرنا مسبقاً

```
>> 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
```

لنقوم باستبدال العنصر في الصف الثالث والعمود الأول إلى الرقم صفر

```
>> B(3,1)=0
```

```
B =
```

```
1  3  7  8
2  6  5 11
0 14 15 13
```

وإذا أردنا استبدال عدة عناصر، مثلاً نريد أن نستبدل الصف الأول والثاني والعمودين من الأول إلى الثالث بقيمة صفر

```
>> B(1:2,1:3)=0
```

```
B =
```

```
0 0 0 8
0 0 0 11
12 14 15 13
```

### حذف أكثر من عنصر من مصفوفة

لا يقوم الماتلاب بعملية حذف لعنصر واحد فقط في مصفوفة، حيث أنه من غير المعقول حذف عنصر من داخل المصفوفة، وبقية الصف والعمود بهم قيم، ولكن إذا أردت أن تقوم بحذف صف كامل أو عمود كامل فيمكن ذلك بعمل التالي. نقوم أولاً بعمل مصفوفة للعمل عليها.

```
>> 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
```

لنقل اننا نريد حذف الصف الثالث كله

وضع أقواس مربعة فارغة تعني عملية حذف

```
>> B(3,:)=[]
```

في خانة الأعمدة تم وضع ( : ) حيث تعني إختيار جميع الأعمدة

الصف الثالث

```
B =
```

```
1 3 7 8
2 6 5 11
```

ولحذف العمود الرابع كله، قم بعمل التالي

```
>> B(:,4)=[]
```

```
B =
```

```
1 3 7
2 6 5
12 14 15
```

### نداء عنصر من مصفوفة

عملية نداء عنصر من أثير العمليات هامة جداً داخل الماتلاب أي أنه نود الحصول على عنصر وحيد من المصفوفة، وذلك بذكر رقم الصف ورقم العمود الذي به هذا العنصر، ولتوضيح هذا الأمر، نقوم بعمل مثال بسيط، معتمدين على نفس المصفوفة التي تم ذكرها في المثال السابق

```
>> 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
```

لنقل اننا نريد العنصر في الصف الأول والعمود الثالث

```
>> B(1,3)
```

```
ans =
```

```
7
```

ولنداء أكثر من عنصر، نقوم مثلاً ببدء الصف الثاني ومن العمود الثاني إلى الرابع

```
>> B(2,2:4)
```

```
ans =
```

```
6  5 11
```

هذا في حالة أننا نعرف حجم المصفوفة، ولكن ماذا إذا لم نكن نعرف حجمها، ونريد أن نحصل على العنصر الأخير مثلاً من الصف الثاني

```
>> B(2,end)
```

```
ans =
```

```
11
```

كلمة end تعني إختيار العنصر

### إيجاد العنصر الأكبر في مصفوفة باستخدام الأمر max

يقوم الماتلاب بإيجاد العنصر الأكبر عن طريق العمل على المصفوفة بشكل مختلف، فكيف يبحث عن العنصر الأكبر في المصفوفة، يقوم الماتلاب بالبحث عن العنصر الأكبر في آل عمود في المصفوفة، وبعدما يقوم بعمل ذلك، يقوم بعمل متجه به الرقم الأكبر من كل عمود، أنظر المثال التالي للتوضيح. لدينا الآن مصفوفة تم إنشائها على الماتلاب.

```
A =
```

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

ولنقم بكتابة الأمر max كما ذكرنا مسبقاً



```
>> B=max(A)
```

```
B =
```

```
23 15 15 11
```

كما تلاحظ فلقد قام الماتلاب باختيار العنصر الأكبر من كل عمود، ولاختيار الرقم الأكبر بينهم يجب كتابة نفس الأمر للنتائج الخارج، وبالتالي نحصل على الرقم الأكبر في المصفوفة ككل

```
>> C=max(B)
```

```
C =
```

```
23
```

### إيجاد العنصر الأصغر في مصفوفة باستخدام الأمر min

هذه العملية أيضاً كثيرة الاستخدام في التطبيقات المختلفة، وهي نفس الخطوات السابق ذكرها في إيجاد العنصر الأكبر ولكن يتم استخدام الأمر min، وإليك المثال التالي

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

```
A =
```

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

```
>> B=min(A)
```

```
B =
```

```
1 1 2 5
```

```
>> C=min(B)
```

```
C =
```

```
1
```

### إيجاد مجموع العناصر في مصفوفة باستخدام الأمر sum

لإيجاد المجموع أما تعلمنا نقوم باستخدام الأمر sum ولكن عملية الجمع يقوم الماتلاب بإيجاد جمع كل عمود على حدى وتوضع في صورة متجه، أما في المثال التالي

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

```
A =
```

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

```
>> B=sum(A)
```

```
B =
```

```
28 21 30 33
```

```
>> C=sum(B)
```

```
C =
```

```
112
```

### إيجاد حاصل ضرب العناصر في مصفوفة باستخدام الأمر prod

يمكن ضرب عناصر المصفوفة، ولكن في الماتلاب عملية الضرب تكون لكل عمود على حدى ويتم وضع الناتج في متجه، وإذا تم استخدام الأمر مرة أخرى يتم ضرب عناصر المتجه جميعها، لينتج حاصل الضرب المصفوفة جميعها، أنظر المثال التالي

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

```
A =
```

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

```
>> B=prod(A)
```

```
B =
```

```
69 60 1080 3850
```

```
>> C=prod(B)
```

```
C =
```

```
1.7214e+010
```

### إيجاد قطر المصفوفة باستخدام الأمر diag

هذه العملية قد تكون ذات استخدام أكاديمي، ولكنها هامة جداً، وخصوصاً أن تلك الخاصية تخدم المصفوفة المربعة (عدد الصفوف يساوي عدد الأعمدة)، ويتم استخدام الأمر diag، وهذا مثال لذلك

```
>> % By defining the Square Matrix A
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]
```

A =

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

```
>> % By Getting the Diagonal of the Matrix A
>> B=diag(A)
```

B =

```
1
1
15
10
```

يمكننا الآن عمل العديد من العمليات على قطر المصفوفة، فمثلاً نريد الحصول على عملية الجمع لعناصر المصفوفة

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

A =

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

```
>> B=sum(diag(A))
```

B =

```
27
```

أو أننا نريد الحصول على حاصل ضرب تلك العناصر

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

```
A =
```

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

```
>> B=prod(diag(A))
```

```
B =
```

```
150
```

### إنشاء المصفوفة السحرية باستخدام الأمر magic

المصفوفة السحرية هي مصفوفة مربعة ذات حيز (n)، مكونة من  $n^2$  أعداد صحيحة، بحيث أن حاصل جمع (n) رقم في كل من الصفوف والأعمدة والأقطار الرئيسية يؤدي لنفس الحاصل.

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↓15

الماتلاب لديه القدرة على إنتاج مصفوفة مربعة يقوم الماتلاب باختيار أرقامها بشكل عشوائي، كل ما عليك هو استخدام الأمر magic وتحديد n.

```
>> A=magic(3)
```

```
A =
```

```
8 1 6
3 5 7
4 9 2
```

```
>> B=magic(9)
```

```
B =
```

```
47 58 69 80 1 12 23 34 45
57 68 79 9 11 22 33 44 46
67 78 8 10 21 32 43 54 56
77 7 18 20 31 42 53 55 66
6 17 19 30 41 52 63 65 76
16 27 29 40 51 62 64 75 5
26 28 39 50 61 72 74 4 15
36 38 49 60 71 73 3 14 25
37 48 59 70 81 2 13 24 35
```

### إيجاد جذور المصفوفة باستخدام الأمر roots

يستخدم الأمر roots لإيجاد حل للمعادلات كثيرة الحدود. يتم تمثيل معاملات كثيرة الحدود في صورة مصفوفة. على سبيل المثال، لإيجاد حل للمعادلة التالية:

$$x^3 + 13x^2 - 18x - 12 = 0$$

نقوم بإنشاء مصفوفة بمعاملات كثيرة الحدود كما يلي:

```
>> t=[1 13 -18 -12]
```

```
t =
```

```
1 13 -18 -12
```

```
>> roots(t)
```

```
ans =
```

```
-14.2075
1.7033
-0.4959
```

وبذلك نحصل ببساطة على قيم  $x$  التي تحقق هذه المعادلة.

### إنشاء مصفوفة ذات قيم تسلسلية

يمكن إنشاء مصفوفة تكون القيم فيها تسلسلية باستخدام الرمز ( : ). على سبيل المثال إذا أردنا إنشاء مصفوفة بفرق 1 بين كل عنصر والعنصر الذي يليه، نقوم بتنفيذ الأمر التالي:

```
>> y=1:5
```

```
y =
```

```
1      2      3      4      5
```

ويمكن تغيير الفارق بوضع الرمز ( : ) يأتي بعدها الفرق.

```
>> y=1:2:10
```

```
y =
```

النهاية:الفرق:البداية

```
1      3      5      7      9
```

ويمكن أيضاً أن يكون الفرق سالباً.

مثال:

في تجربة في مختبر الدوائر الكهربائية يتم زيادة الفولتية تدريجياً من 2 إلى 10 بزيادة 2 فولت. والمقاومة تساوي 260 أوم. اوجد مقدار القدرة المستهلكة في المقاومة في كل مرحلة من مراحل الزيادة.

```
>> v=2:2:10 ;
```

```
>> R=260;
```

```
>> i=v/R;
```

```
>> P=i.*v
```

لاحظ أهمية النقطة قبل إشارة الضرب في السطر الرابع

```
P =
```

```
0.0154    0.0615    0.1385    0.2462    0.3846
```

## الفصل الثالث الملفات M-File

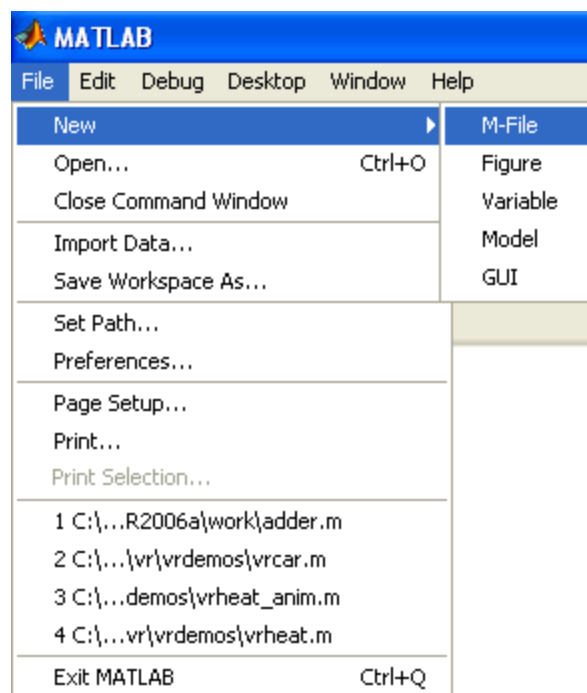
## تعريف M-File

في الفصول الماضية قمنا باستخدام الماتلاب كآلة حاسبة متطورة نوعاً ما، لكن لم تتعدى تطبيقاتنا عن الحسابات وإدخال المعلومات بشكل يدوي أو تعريفها مسبقاً لكن بدون عمليات منطقية. لكن في هذا الفصل الذي يعتبر الأكثر أهمية ومتعة في ذات الوقت سوف نتعرف على لائحة M-file في الماتلاب. التي يشبهها البعض بالبرمجة باستخدام لغة البرمجة C++ لكن باختلاف كبير لأن تطبيقاتها أوسع بكثير وأشمل وأروع.

ما يميز M-File كوسيلة لإدخال الأوامر وليس من خلال نافذة الأوامر Command Window، هو ما يلي:

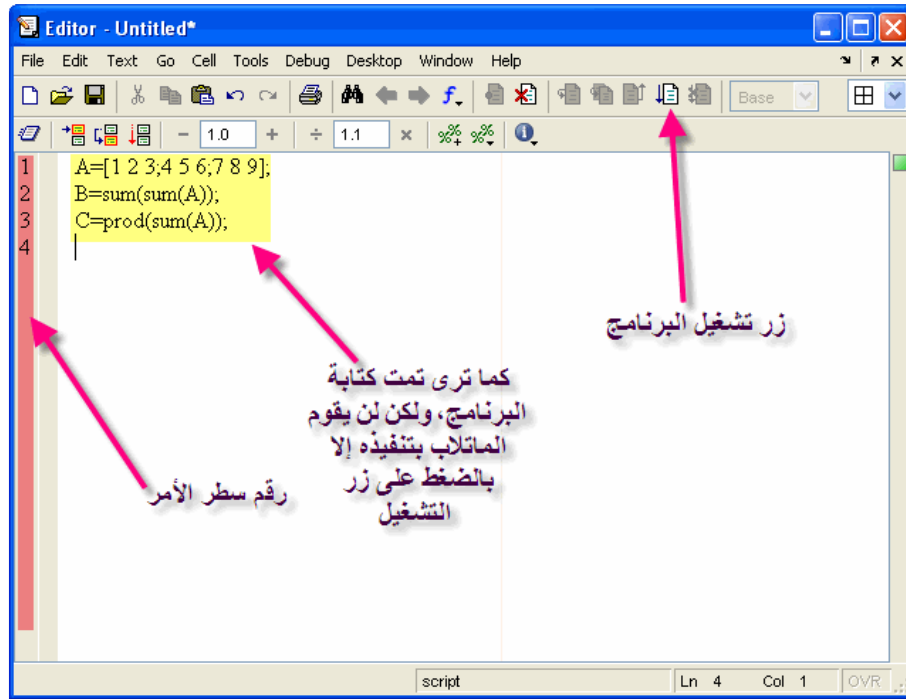
١. في عملية إدخال الأوامر التي كنا نستخدمها، إذا أردنا تعديل عنصر أو أكثر كان يجب إعادة إدخال الأمر من جديد.
٢. إذا وجد خطأ، فيجب كتابة الأمر من جديد.
٣. إذا كتبنا برنامج كبير، وأردنا إعادة العملية مرة أخرى يجب إدخال جميع الأوامر من جديد وبنفس الترتيب.
٤. إذا حدث خطأ في ترتيب الأوامر لهذا البرنامج الكبير ستقوم بإعادة الإدخال الأوامر من البداية مرة أخرى.
٥. يصعب عمل عملية تصحيح للأخطاء Debugging

وهذا بالطبع يستغرق وقتاً كبيراً هذا بالإضافة إلى الملل الذي يحدث للمستخدم وحلاً لهذه المشكلة، تم عمل ما يسمى M-File والتي تعطي القدرة على كتابة البرنامج كاملاً أولاً بدون تشغيل، وبعد الانتهاء منه يتم تشغيله، هذه الخاصية تعطي القدرة على تعديل القيم دون الحاجة إلى كتابتها مرة أخرى، أو إعادة إدخال الأوامر التي تعتمد على هذا الأمر. فكيف يتم تشغيل تلك الخاصية؟ اتبع الصورة التالية.



وبالتالي ستظهر نافذه جديدة، سنقوم الآن بالتعرف على نافذة M-File.



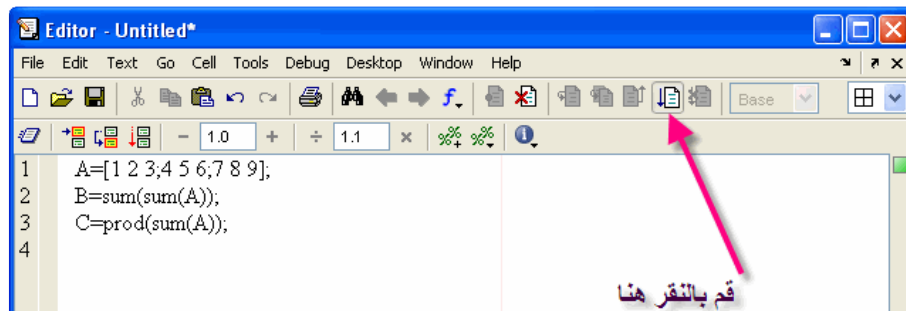


عند الضغط على زر التشغيل، سيطلبك الماتلاب بحفظ البرنامج، ولكن يشترط الآتي عند حفظ البرنامج.

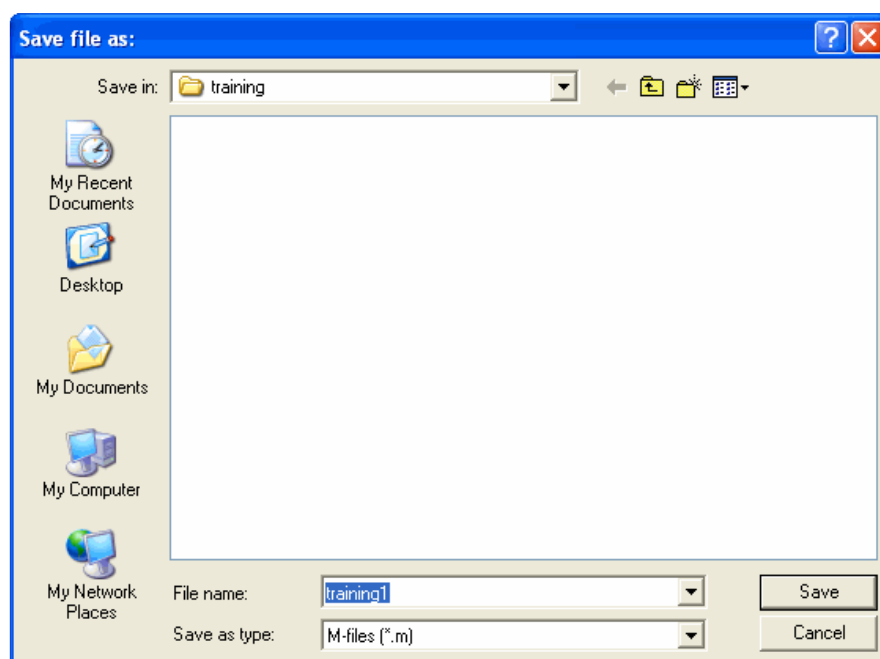
١. ألا يبدأ اسم الملف بأرقام
٢. ألا يكون اسم الملف أمراً معروفاً في الماتلاب
٣. ألا يحتوي اسم الملف على مسافات فاصلة
٤. ألا يحتوي اسم الملف على رموز خاصة مثل \*، &، +، -

يجب مراعاة تلك الشروط وإلا لن يقوم الماتلاب بتنفيذ البرنامج فالنقم بتنفيذ المثال المكتوب الآن في النافذة السابقة.

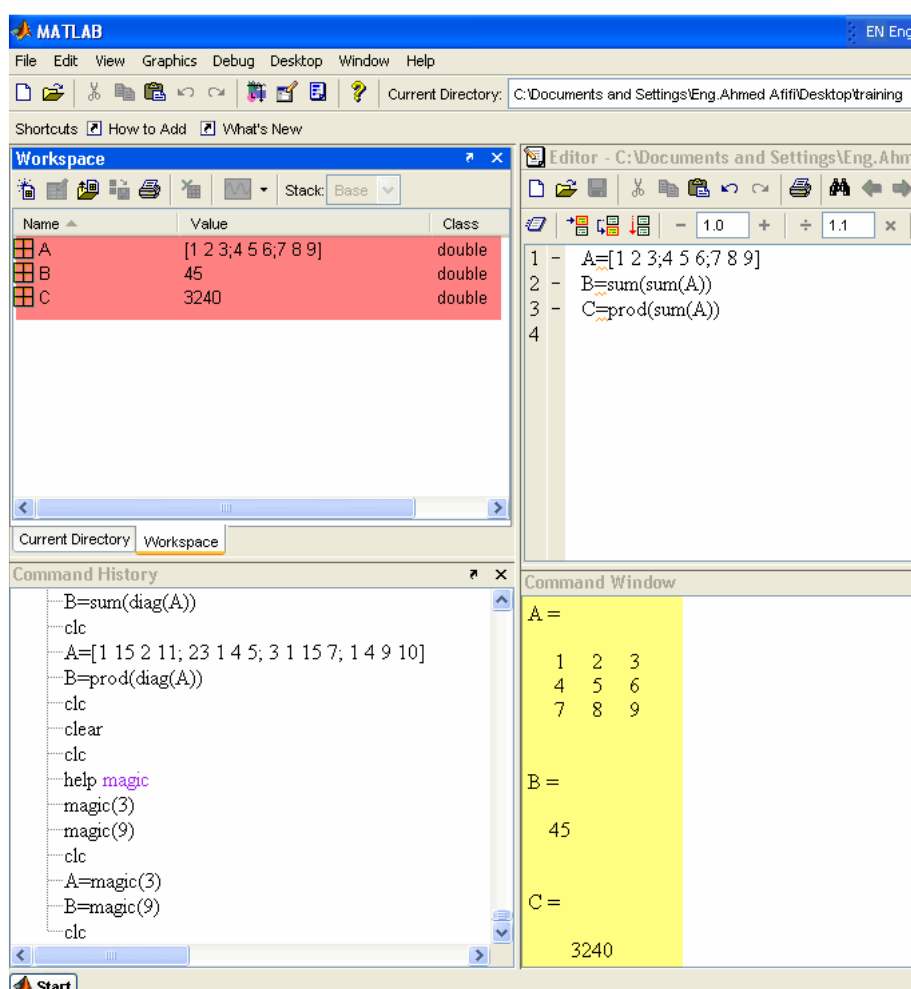
١ - يتم الضغط على زر التشغيل كما هو واضح في الصورة التالية.



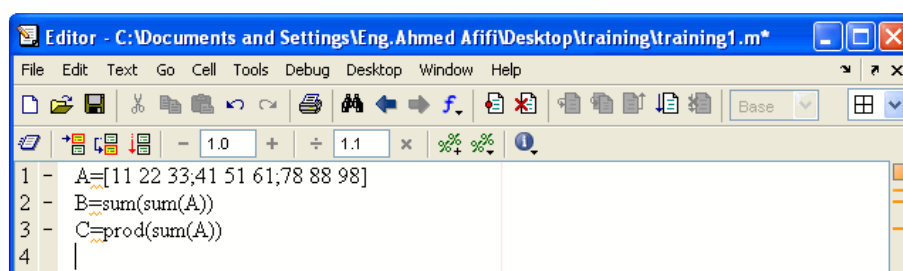
٢ - سيطلبنا الماتلاب بحفظ البرنامج أولاً، ولنسميه training1.



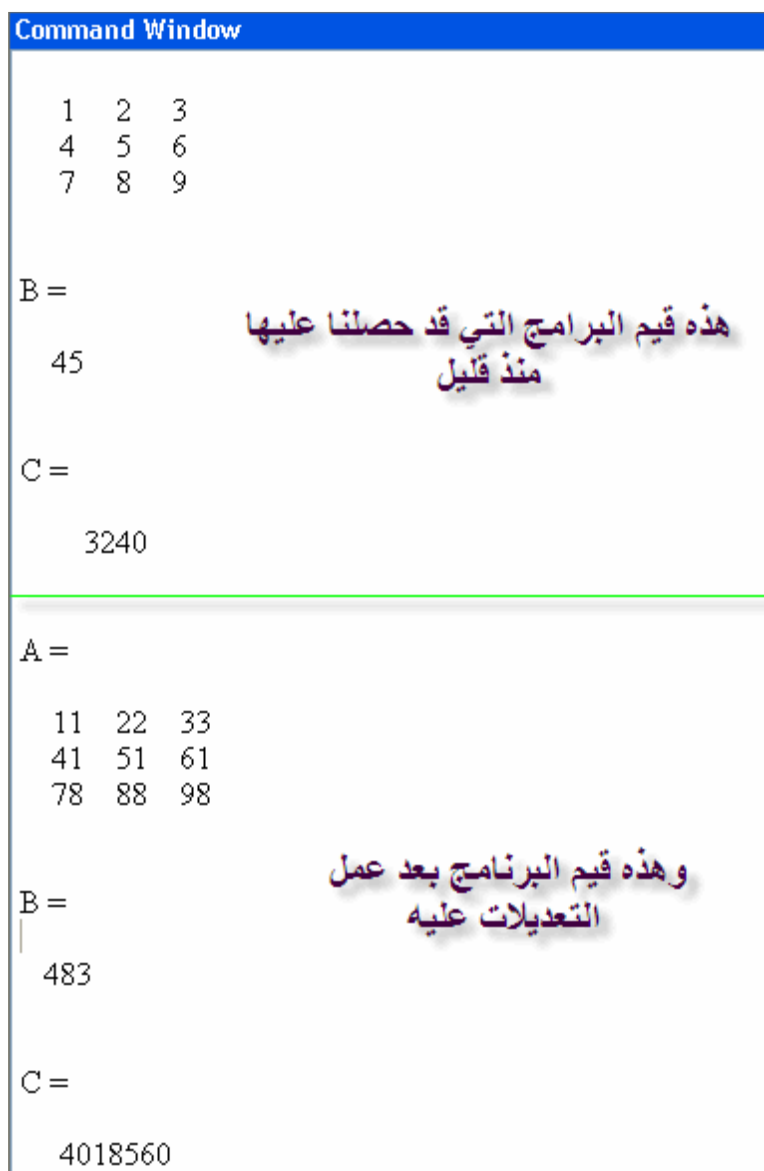
٣ - ستظهر القيم في كلاً من Command Window and Workspace.



٤ - لنعود إلى M-File ونقوم بتغيير بعض القيم للمصفوفة، كما في الشكل التالي.



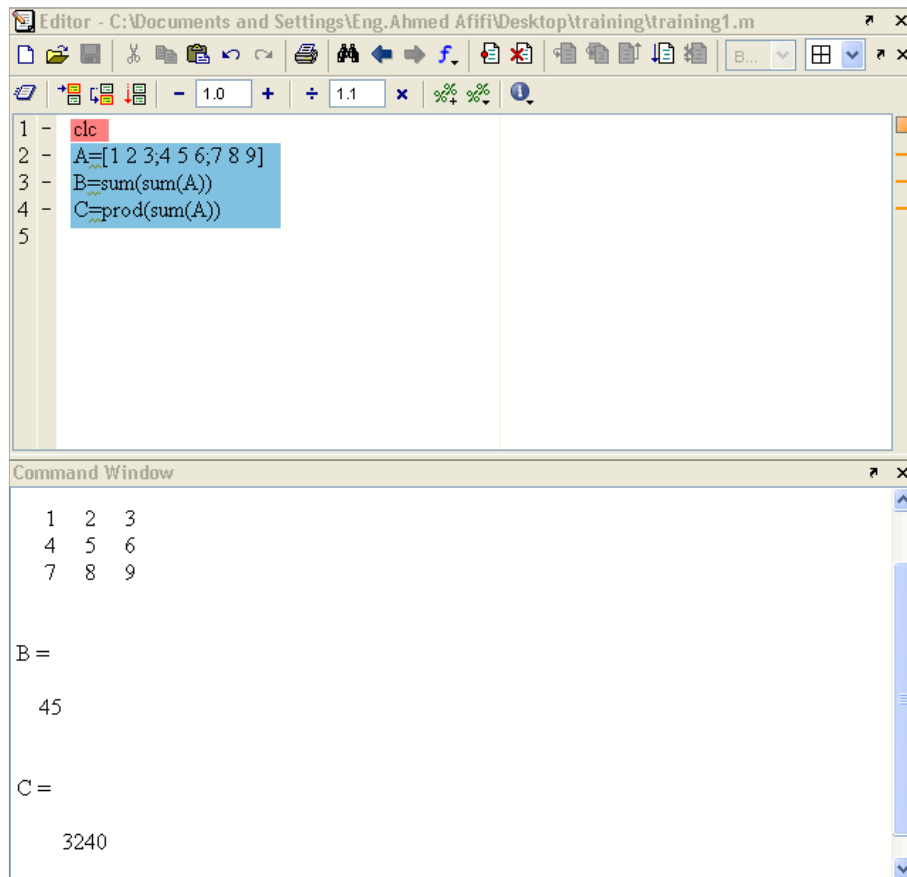
٥ - سنقوم الآن بتشغيل البرنامج، وسيقوم الماتلاب الآن بالحفظ تلقائياً دون الحاجة لإعادة التسمية، ثم شاهد نافذة الأوامر Command Window.



## الأمران CLC و Clear

وكما تلاحظ فإنه في كل عملية تحديث للبرنامج ستظل قيم البرنامج القديم موجودة، فحلاً لهذه المشكلة، يتم وضع الأمر CLC في أول كل برنامج، وهذا سيكون مبدأ في جميع البرامج التي نقوم بعملها، لا بد من أن تبدأ بهذا الأمر.

## سنقوم الآن بكتابة M-File



سنقوم الآن بتعديل المثال، وحتى نتأكد أن أمر CLC يعمل، ستختفي القيم من Command Window وتظهر القيم الجديدة.

```

1 - clc
2 - A=[11 21 31;42 52 62;73 38 39]
3 - B=sum(sum(A))
4 - C=prod(sum(A))
5

```

Command Window

```

11 21 31
42 52 62
73 38 39

B =

369

C =

1846152

```

كما ترى فإن القيم السابقة اختفت وظهرت القيم الجديدة

لنقم بتعديل بسيط في البرنامج عن طريق تغيير الرموز فقط من A، B، C إلى D، E، F ومشاهدة النافذة .Workspace

```

1 - clc
2 - D=[1 2 3;4 5 6;7 8 9];
3 - E=sum(sum(D));
4 - F=prod(sum(D));
5

```

Workspace

Name	Value	Class
A	[1 2 3;4 5 6;7 8 9]	double
B	45	double
C	3240	double
D	[1 2 3;4 5 6;7 8 9]	double
E	45	double
F	3240	double

قيم البرنامج الأول قبل تغيير الرموز لا تزال موجودة

ولتلافي هذه المشكلة، يجب وضع أمر Clear بعد الأمر CLC بحيث يقوم بمسح أي قيمة سابقة من أي برنامج آخر في Workspace ويجب تثبيت هذا الأمر أيضاً في جميع البرامج والتي سيتم عملها لاحقاً. وسنقوم الآن بتنفيذ نفس البرنامج ولكن بعد وضع الأمر Clear ونلاحظ الفرق.

```

1 - clc
2 - clear
3 - D=[1 2 3;4 5 6;7 8 9];
4 - E=sum(sum(D));
5 - F=prod(sum(D));
6

```

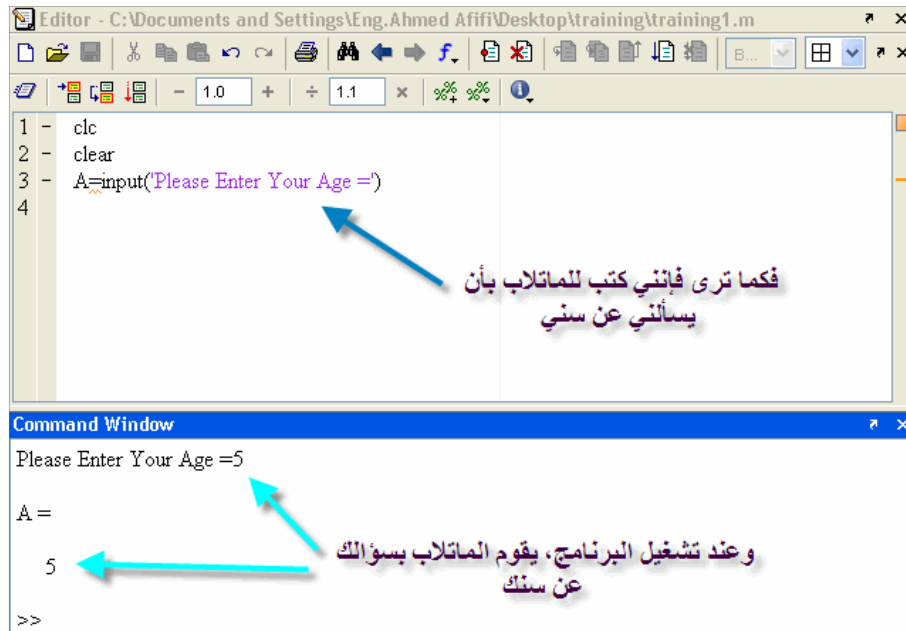
Workspace

Name	Value	Class
D	[1 2 3;4 5 6;7 8 9]	double
E	45	double
F	3240	double

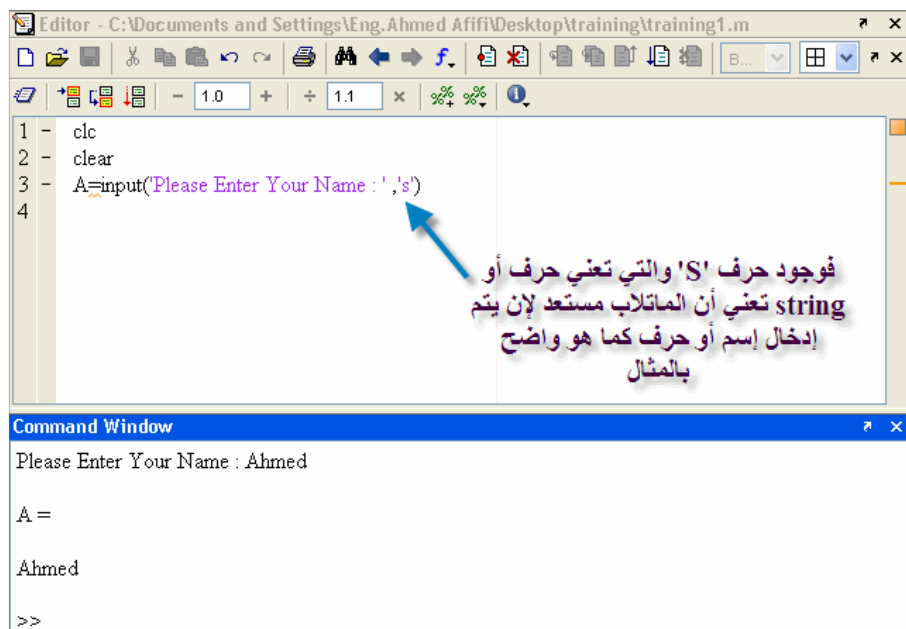
كما ترى اختفت الرموز القديمة باستخدام الأمر clear

## الأمر Input

هذا الأمر هام جداً في الماتلاب، حيث يقوم المستخدم من خلال استخدامه بجعل الماتلاب يسأله عن متجه أو مصفوفة، أو حتى حروف وأسماء وذلك تبعاً لما يقوم المستخدم بتعريفه فمثلاً إذا أردنا الماتلاب يطالبك بإدخال سنك، سنقوم بعمل التالي

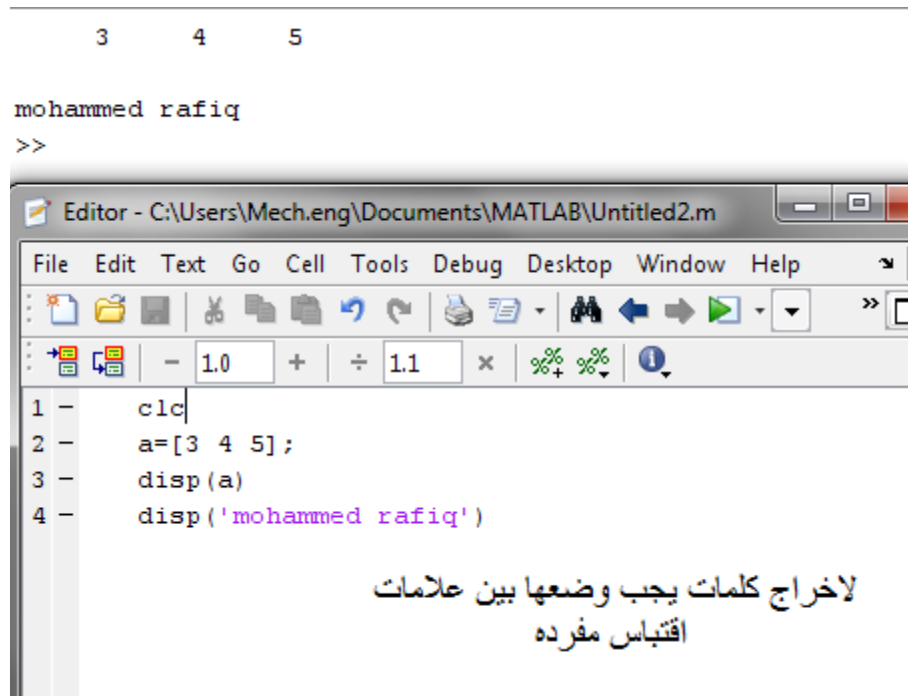


وإذا أردنا أن نعرف للماتلاب بأن ما سيتم إدخاله هو اسم وليس رقم، يتم كتابة البرنامج بالشكل التالي



## الأمر disp

هذا الامر يساوي cout في لغة البرمجة C++ ويعني إظهار أو إخراج.



### التحويل بين string و number

هناك نوعان من البيانات الأولى string والثانية number. حيث أن الماتلاب يعامل الأول كأنه رمز وعندما يتم جمعه أو طرحه مع رقم ثان يعطي نتيجة خاطئة، أما الثاني فهو رقم ويعامل معاملة الرقم.

لتحويل string الى number والعكس نستخدم احد الأمرين str2num أو num2str.

String: هو حرف أو كلمة في الماتلاب وهو يستخدم مع البيانات النصية.

number: هو رقم أو مجموعة من الأرقام.

سنقوم بمثال بسيط حتى تفهم المقصود من هذه الأوامر، سنقوم باستخدام الأمر input في وضعية string وسنقوم بإدخال أرقام.

1 - clc  
2 - clear  
3 - age=input('Please Enter Your Age ','s')

Command Window  
Please Enter Your Age 5  
age =  
5  
>> check=2\*age  
check =  
106  
>>

١ - تم استخدام الأمر input في صورة character وليس string  
٢ - قمنا بإدخال رقماً للتعريف على أنه سن الإنسان، ولكن هل تعرف الماتلاب على أنه رقم  
٣ - في الحقيقة لم يتعرف الماتلاب على أنه رقم بل تعرف على أنه string وهذا نتيجة استخدام الأمر input في وضعية string

سنقوم الآن باستخدام str2num لتحويل string إلى number.

1 - clc  
2 - clear  
3 - age=input('Please Enter Your Age ','s');  
4 - age\_modified=str2num(age)

Command Window  
Please Enter Your Age 5  
age\_modified =  
5  
>> check=2\*age\_modified  
check =  
10  
>>

في هذه المرة تعرف الماتلاب على أن الذي يتم إدخاله هو رقم وهذا واضح من خلال عملية الضرب في ٢ كعملية تأكيد، وذلك بسبب استخدام الأمر str2num

أما بخصوص الأمر الثاني num2str فيستخدم في تحويل الأرقام إلى string فالمثال التالي يطلب منا إدخال الاسم والسن، ثم وضع الاسم والسن في متجه، ولكن يجب أن نراعي أن المتجه إما يحتوي على أرقام أو string فقط وليس كلاهما.



The screenshot shows the MATLAB Editor window with a script named 'training1.m'. The script contains the following code:

```

1 - clc
2 - clear
3 - name=input('Please Enter Your Name ','s');
4 - age=input('Please Enter Your Age = ');
5 - answer=['Your name is ',name,' and you are ',num2str(age),' Years Old']

```

Below the code, there is a comment in Arabic:

٢- وهذه هي صورة المتجه، ولكن كما ترى تم وضعها في صورة string مما يتطلب تحويل الرقم age إلى string باستخدام num2str

The Command Window shows the execution of the script:

```

Please Enter Your Name Ahmed
Please Enter Your Age = 22

answer =

Your name is Ahmed and you are 22 Years Old

```

At the bottom of the Command Window, there is another comment in Arabic:

١- لإظهار الكتابة بهذا الشكل، لا بد من وضعها في صورة متجه

## الفصل الرابع الجمل الشرطية والبرمجة

## المقارنة

فالجمل الدورانية أو الشرطية مشابهة تماماً لما في برمجة اللغة C++ مع اختلاف بسيط وسيتم توضيحها في هذا الفصل. يمكن إجراء هذه العمليات على عناصر منفردة أو مصفوفات.

الوظيفة	الرمز
أقل من	<
أكبر من	>
أقل من أو يساوي	<=
أكبر من أو يساوي	>=
هل يساوي؟	==
لا يساوي	~=
الدالة المنطقية AND	&
الدالة المنطقية OR	!
الدالة المنطقية NOT	~

شاهد المثال

```
>> x=[3 4 5];
y=[3 9 1];
z=(x==y)
```

z =

1 0 0

```
>> z=x>y
```

z =

0 0 1

```
>> z= x<=y
```

z =

1 1 0

ياخذ كل عنصر على حدى  
للمقارنة

وهذا المثال أيضاً

```
x =
     2     3     4
     2     4     5
    665     6     4

>> y=[3 4 5 ; 6 3 4 ; 13 5 4 ]

y =
     3     4     5
     6     3     4
    13     5     4

>> z=x (y>x)
معنى هذا الأمر هو إيجاد عناصر في
x تحقق الشرط
y>x
z =
     2
     2
     3
     4
```

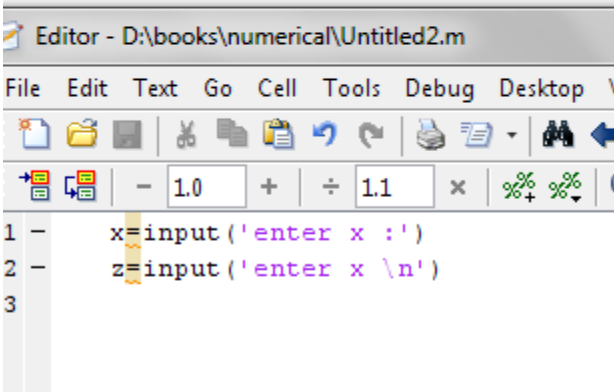
### الأمر \n

تستطيع وضع آخر جملة طلب الإدخال الرمز \n حتى يتم طلب الإدخال بنزول سطر.

```
enter x :5

x =
     5

enter x
5
```



دون استخدام \n قام البرنامج بطلب قيمة x بنفس السطر، اما بوجودها طلبها بسطر جديد.

## الأمر If

هناك ثلاث أنواع لجملـة If:

- simple if statement
- nested if statement
- if-else statement

### simple if statement

هذا النظام فيه جملة If واحده فهو نظام بسيط. الشكل العام هو كما يلي.

```
if expression 1
    statement group 1
end
```

### nested if statement

هذا النظام فيه أكثر من جملة If داخل بعضها (لا ينظر بتحقيق الشرط الثاني الا إذا تحقق الأول). الشكل العام هو كما يلي.

```
if expression 1
    statement group 1
    if expression 2
        statement group 2
    end
    statement group 3
end
statement group 4
```

### if-else statement

هذا النظام فيه ينتقل عند عدم تحقيق الشرط إلى else ويحققها. الشكل العام هو كما يلي.

```
if expression 1
    statement group 1
else
    statement group 2
end
```

مثال ١:

اكتب برنامج يقوم بحساب الجذر للأرقام الموجبة فقط.

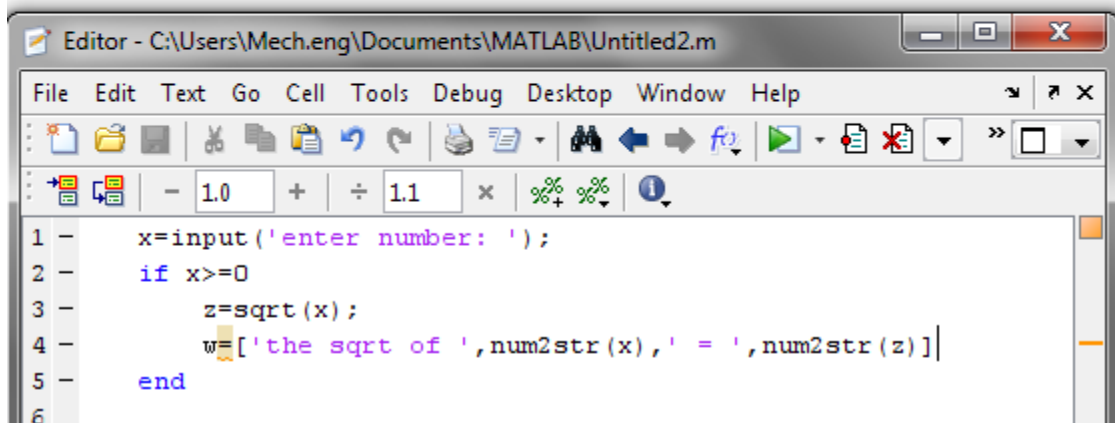
الحل: هنا يوجد شرط واحد فقط. إذا لم يتحقق لن يتم تحقيق أي شيء.

```
enter number: -15
enter number: 25
```

```
w =
```

```
the sqrt of 25 = 5
```

```
>>
```



لاحظ اننا هنا استخدمنا الأمر num2str لأنه كما ورد سابقاً المصفوفة يجب ان تكون كلها string أو كلها numbers والمصفوفة W في السطر الرابع في مثالنا نريدها أن تكون مصفوفة رموز strings فحولنا x و z الى حروف. لو قمنا بتحويل x من البداية الى حروف بأن نكتب هذا الأمر

```
x=input('enter number','s')
```

سوف يتم تعريف x على اساس انها رمز فعند إيجاد جذرها لن يتم إيجاده بالطريقة الصحيحة.

مثال ٢:

اكتب برنامج يقوم بإيجاد الجذر للأرقام الموجبة فقط وإذا كان جوابه كسراً يقوم بكتابة good luck وإذا كان الجواب ليس كسراً يقوم بإخراج الجواب.

الحل: هناك شرطان، الشرط الثاني لا يتحقق الا بتحقيق الشرط الأول وفي هذه الحالة يتم استخدام النمط الثاني من جمل If.

enter number: 25

z =

5

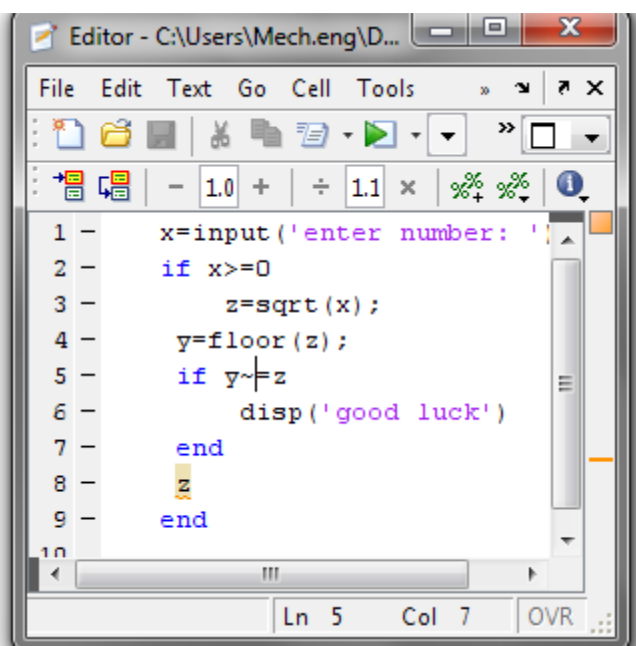
enter number: 26

good luck

z =

5.0990

>>



لاحظ استخدام الأمر floor الذي عني أكبر عدد صحيح.

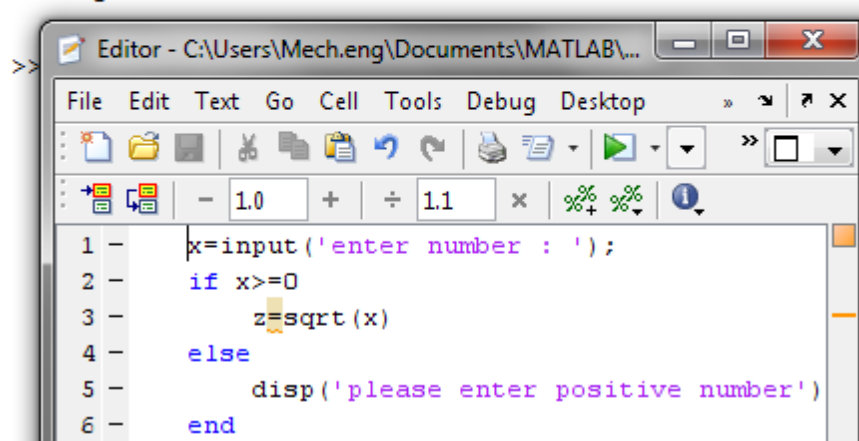
مثال ٣: اكتب برنامج يقوم بإيجاد جذر الأرقام الموجبة وإذا كان الرقم سالباً يقوم بكتابة please enter positive number.

الحل: هنا يوجد شرطان إذا لم يتحقق الأول سيتم يتحقق الثاني اذا نستخدم النظام الثالث.

enter number : -35  
please enter positive number  
enter number : 25

z =

5



مثال ٤:

اكتب برنامج يطلب إدخال المقاومات و ثم يسأل إذا كانوا على توازي أم توالي ويتم إيجادهم.

الحل: هنا يوجد شرطان إذا لم يتحقق الأول سيتم تحقيق الثاني إذاً نستخدم النظام الثالث. لكن هنا يتم طلب إما موازي أو توالي. ماذا إذا أدخل المستخدم كلمة غير متوقعة؟  
هنا نستخدم نظام شبيه لنظام else لكان بأكثر من else. انظر الجواب وتمعنه جيداً.

```

enter the resistance: [3 6]

R =

     3     6

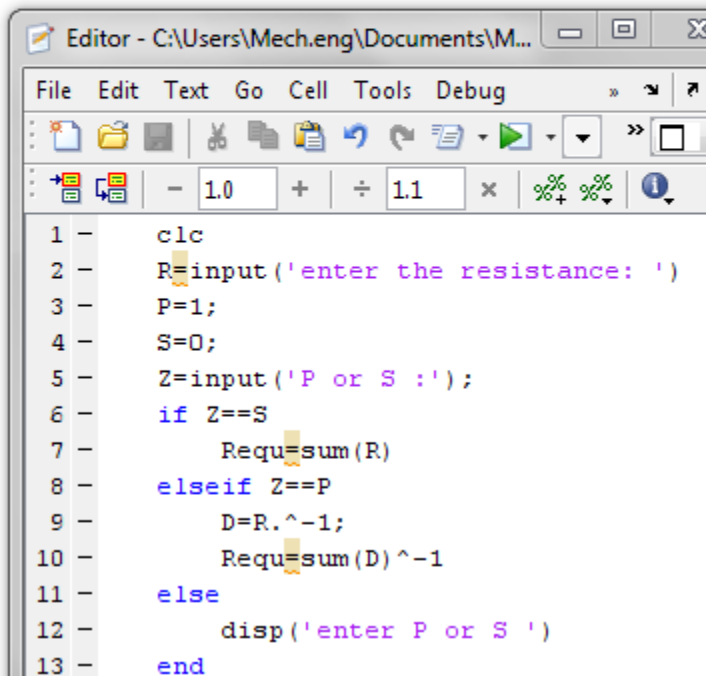
P or S :P

Requ =

     2

>>

```



```

1 - clc
2 - R=input('enter the resistance: ')
3 - P=1;
4 - S=0;
5 - Z=input('P or S :');
6 - if Z==S
7 -     Requ=sum(R)
8 - elseif Z==P
9 -     D=R.^-1;
10 -    Requ=sum(D)^-1
11 - else
12 -     disp('enter P or S ')
13 - end

```

## For الأمر

الشكل العام يكون كما يلي.

```

for i:s:f
    statement group X
end

```

جملة for هي أحدي الجمل الدورانية المهمة في عالم الماتلاب.

i: تعني القيمة الابتدائية وممكن أن تكون متغير.

s: تعني الخطوة وممكن أن تكون متغير وهي ليست إجبارياً كتابتها. إذا لم يتم كتابتها سيأخذ الخطوة تساوي 1.

f: القيمة النهائية.

لطالما جملة for بين القيمة الابتدائية والنهائية سيتم تنفيذ البرنامج، عندما تخرج عن النطاق سيذهب الى end.



مثال:

اكتب برنامج لحساب المضروب (factorial).

الحل:

لإيجاد المضروب بالماتلاب نستخدم الأمر التالي factorial(var or num).

enter number 6

x =

6

z =

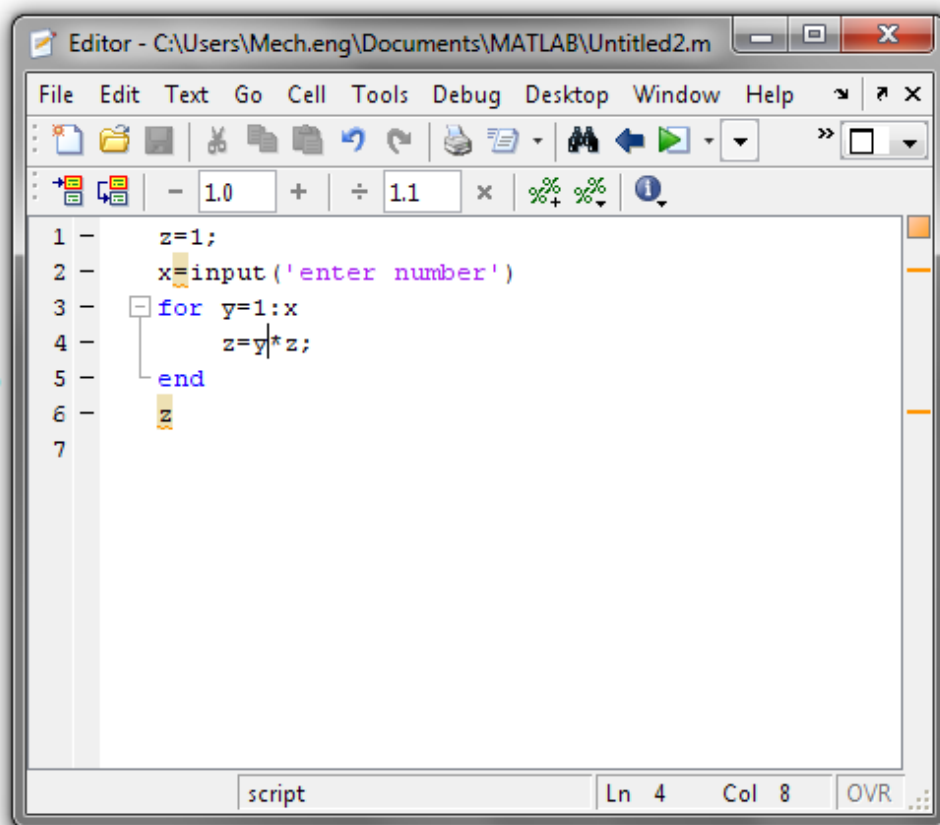
720

>> factorial(6)

ans =

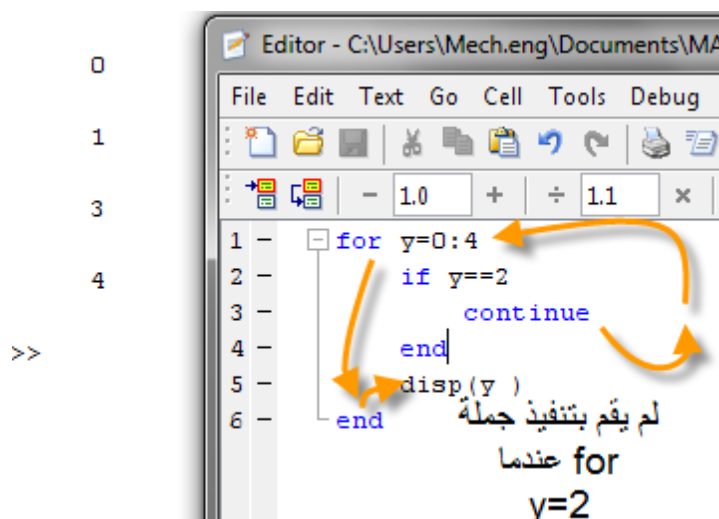
720

>>

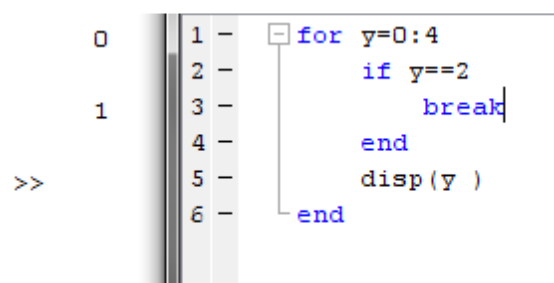


## الأمرين Break و Continue

يستخدم هذان الأمران للتخطي أو الإيقاف. فالأمر Break عند تحقق الشرط يقوم بإيقاف الحلقة Loop والأمر Continue عند تحقق الشرط يقوم بالتخطي والعودة لجملة If.



وتوضيح للأمر Break



عمل الأمر Break على توقيف جملة For.

مثال:

اكتب برنامج يرسم هذه المعادلة.

$$y = \begin{cases} 15\sqrt{4x+10} & x \geq 9 \\ 10x+10 & 0 \leq x \leq 9 \\ 10 & x < 0 \end{cases}$$

للقيم  $-5 \leq x \leq 30$

الحل:

```
for x=-5:30;
    if x>=9
        y=15*sqrt(4*x)+10;
    elseif x>=0
        y=10*x+10;
    else
        y=10;
    end
    q(x+6)=y;
end
```

```
r(x+6)=x;
end
q
r
plot(r,q)
```

مثال:

اكتب برنامج يقوم بحسب أول 15 لهذه المتسلسلة.

$$5k^2 - 2k$$

### الأمـر While

الشكل العام هو كما يلي.

```
while expression 1
statement group 1
end
statement group 2
```

وهذه الجملة تعني أنه طالما الشرط expression 1 متحقق سيتم تحقيق statement group 1 وعندما لا يتحقق سوف يتم تحقيق statement group 2.

## الفصل الخامس الرسم

## الرسم ثنائي الأبعاد 2D Plotting

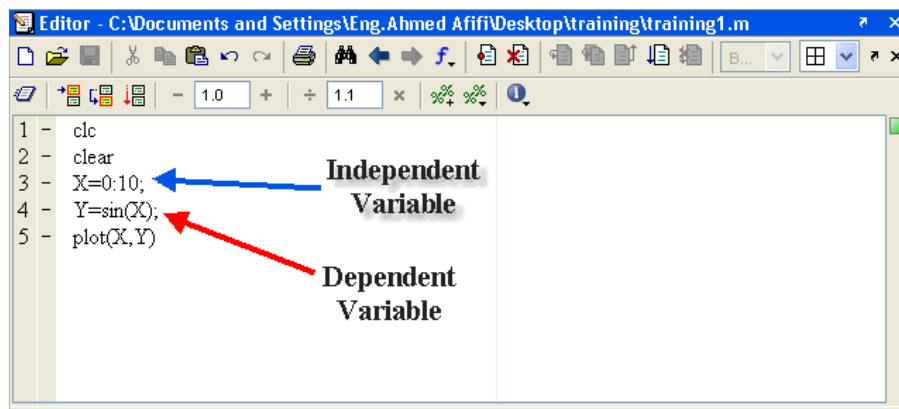
المقصود بالرسم ثنائي الأبعاد هو ان تكون العلاقة التي تحكم عملية الرسم تكون بين متغيرين فقط أحدهما يسمى independent variable والآخر يسمى dependent. المتغير independent Variable هو المتغير المستقل أي أن قيمه لا تحكمها علاقة، بينما المتغير dependent Variable هو المتغير المعتمد حيث يعتمد قيمه على قيم المتغير المستقل.

### الأمر Plot

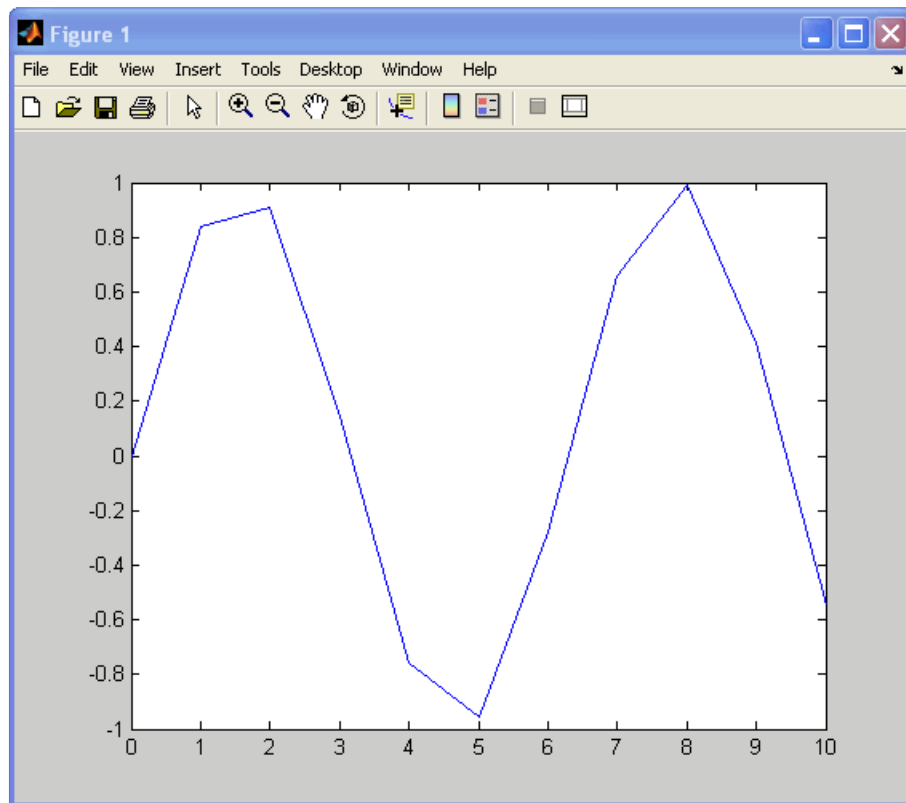
أمر الرسم Plot يأخذ الصورة التالية.

Plot(independent variable, dependent variable)

وهذا مثال بسيط لكيفية رسم Sine Wave



وتلاحظ أننا قد اخترنا 10 نقاط فقط لرسم Sine Wave، وهذا عدد قليل لرسم Sine Wave وتلاحظ ظهور الرسمة بالشكل التالي.



وحلاً لهذه المشكلة، لابد من زيادة عدد النقاط داخل المتجه، كما في الشكل التالي

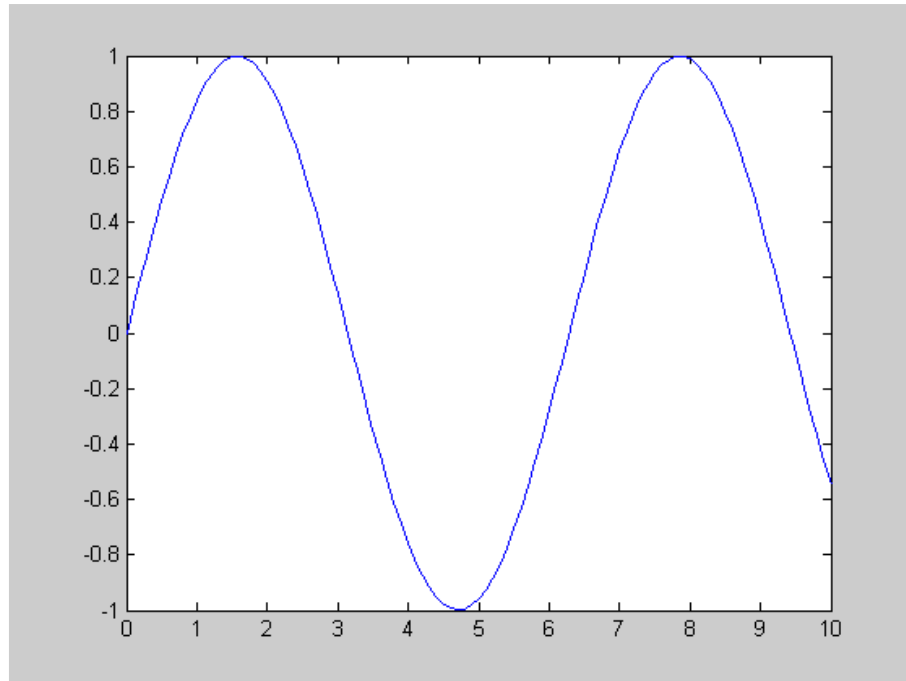
```

1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y)
6

```

يمكن وضع المتجه في هذه الصورة  
minimum number : step : maximum number

وستلاحظ أن الرسمة قد تحسنت كثيراً



### عملية وضع شبكة على الرسم

يقوم الماتلاب بوضع شبكة على الرسم، بحيث يكون من السهل تحديد القيم من على الرسم حيث تأخذ الأمر grid بعد الأمر plot.

### إضافة خصائص إلى الرسومات

في بعض الأحيان يكون من الضروري جداً تغيير بعض الخواص لدى الرسومات التي نحصل عليها مثل تغيير الألوان، وتغيير الرزمة من خطوط متصلة إلى نجوم ونقاط وغيرها، وهذه هي مجموعة الخصائص التي تتم من خلال الماتلاب.

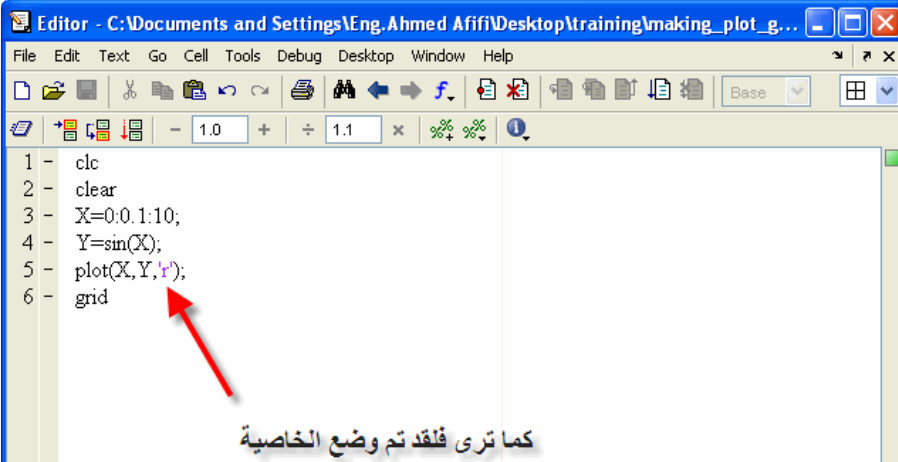
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-. dashdot	
c	cyan	+	plus	-- dashed	
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

تكون هذه الخصائص متضمنة في الأمر plot حيث تأخذ الصورة التالية.

`plot( independent Variable, Dependent Variable, ' the property ' )`

كما ترى فإن أي خاصية يتم وضعها  
بعد **Dependent Variable**  
ولكن يجب وضع الخاصية بين  
فاصلتين 'الخاصية'

اعتماداً على المثال السابق سنقوم بتعديل بعض الخصائص سنقوم، مثلاً بتغيير لون الخط إلى الأحمر.

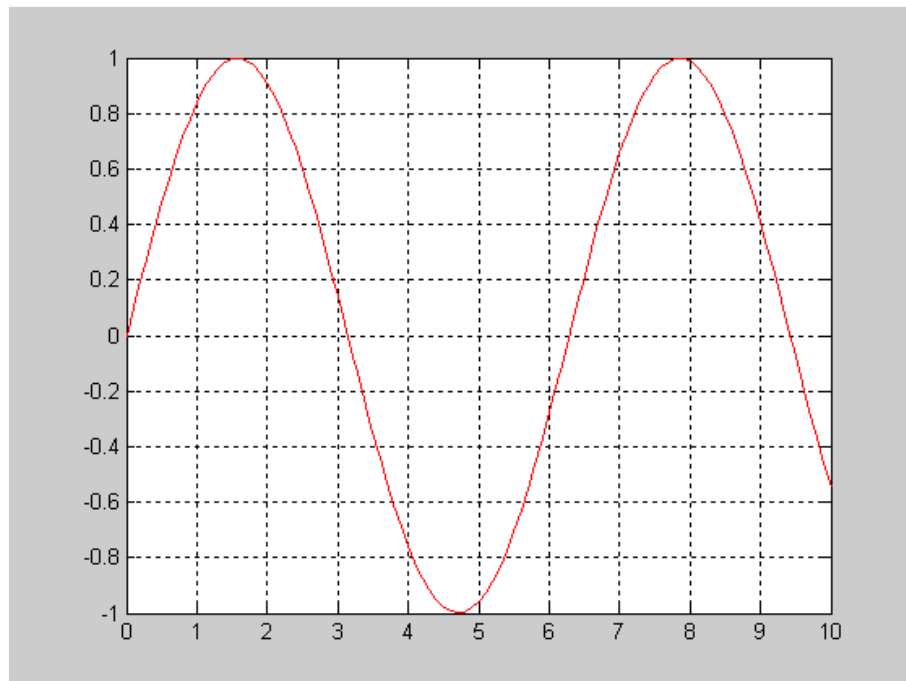


```

1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y,'r');
6 - grid
    
```

كما ترى فلقد تم وضع الخاصية

وبالتالي نحصل على الصورة التالية



ولنقم الآن بإضافة خاصية جديدة بأن يكون الخط ليس خطاً متصل وإنما عبارة عن نجوم

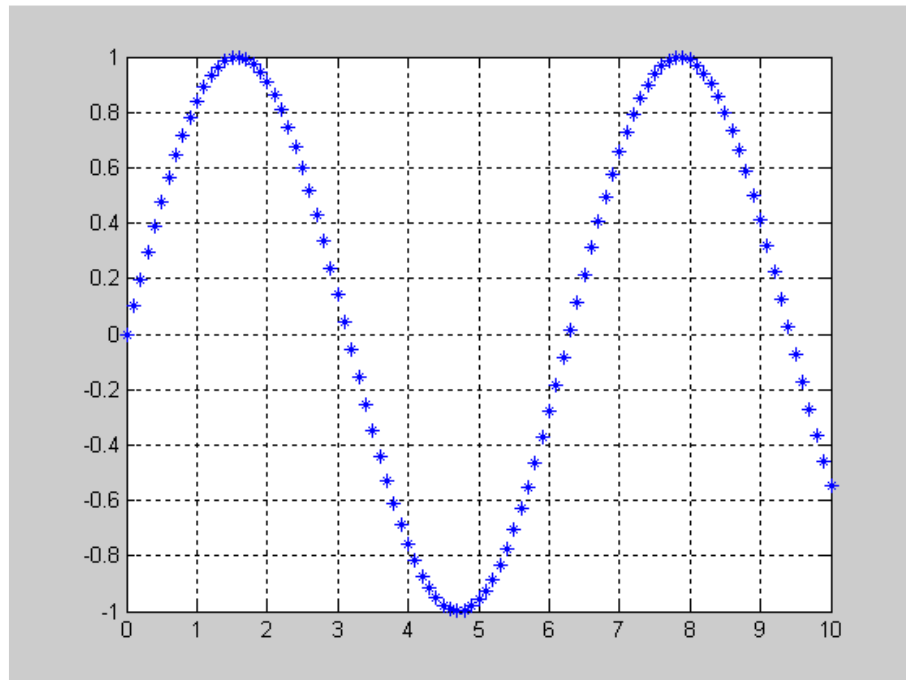


```

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y,'*');
6 - grid

```

وبالتالي نحصل على الشكل التالي



وإذا أردنا أن نحصل على نجوم حمراء (أي دمج الخاصيتين معاً)

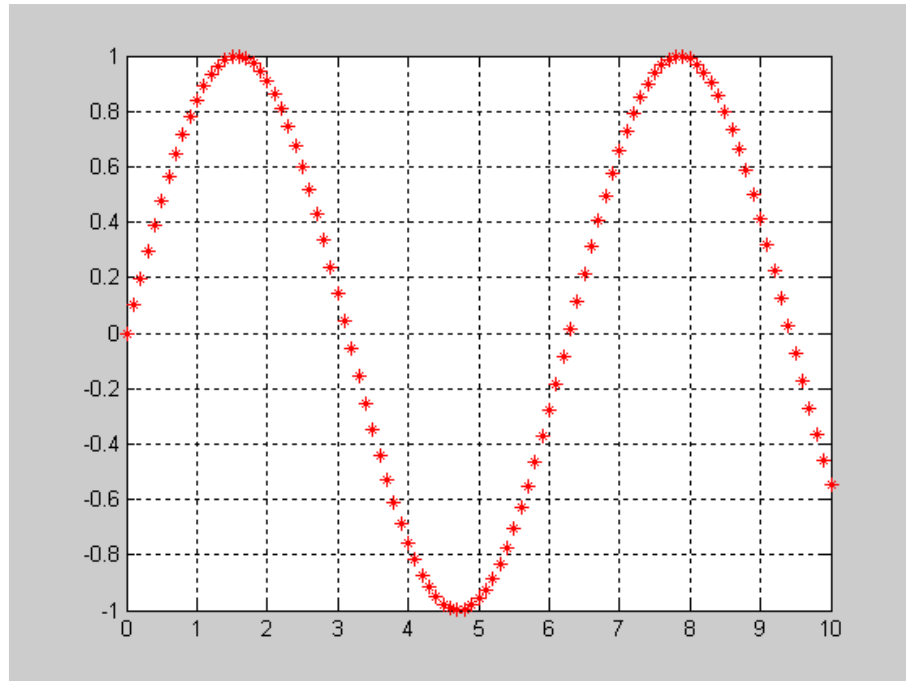
```

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y,'r*');
6 - grid
7 -

```

تم دمج الخاصيتين معاً، بذكر اللون ثم شكل الخط

وبالتالي نحصل على الشكل التالي



الآن سنقوم بعمل معادلة آخر بالإضافة إلى المعادلة المذكورة بحيث يكون لدينا رسمتان, بحيث تأخذ الشكل التالي

```

1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - Z=cos(X);
6 - plot(X,Y)
7 - plot(X,Z)
8 - grid

```

ولكن عند تشغيل البرنامج، سيقوم الماتلاب بإظهار الرسمة الأخيرة فقط، فكيف يتم إظهار الرسمتين، يتم ذلك باستخدام الأمر Hold on قبل الأمر plot لكي يتم وضع الرسمتين في نافذة واحدة، وفي نهاية الأمر يتم وضع الأمر hold off.

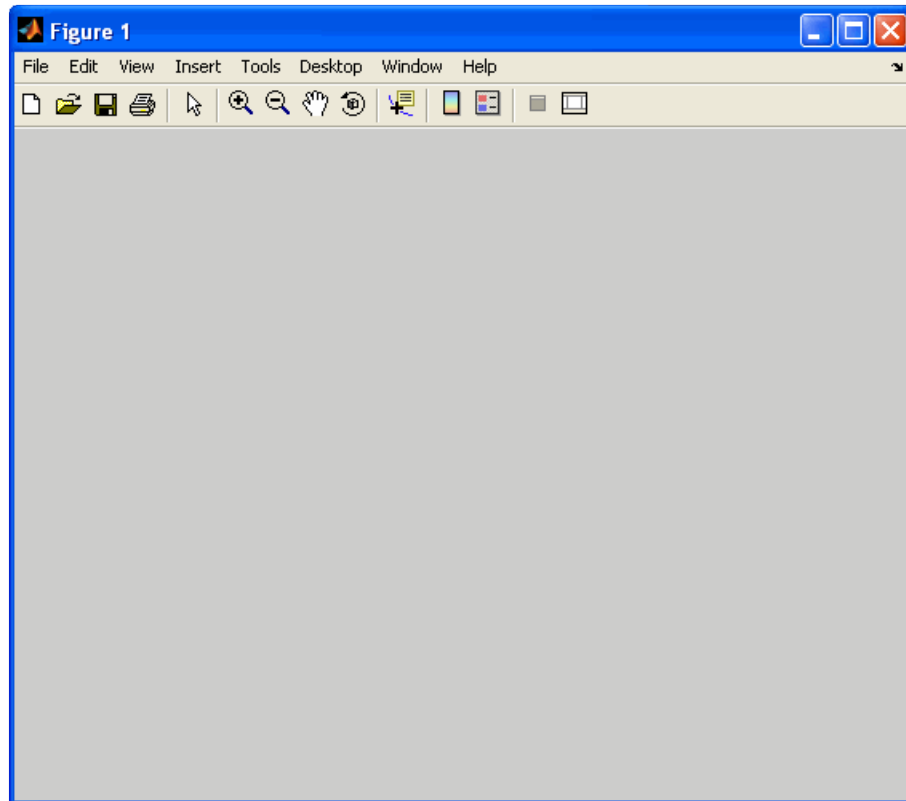
```

1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - Z=cos(X);
6 - hold on
7 - plot(X,Y)
8 - plot(X,Z)
9 - grid
10 - hold off

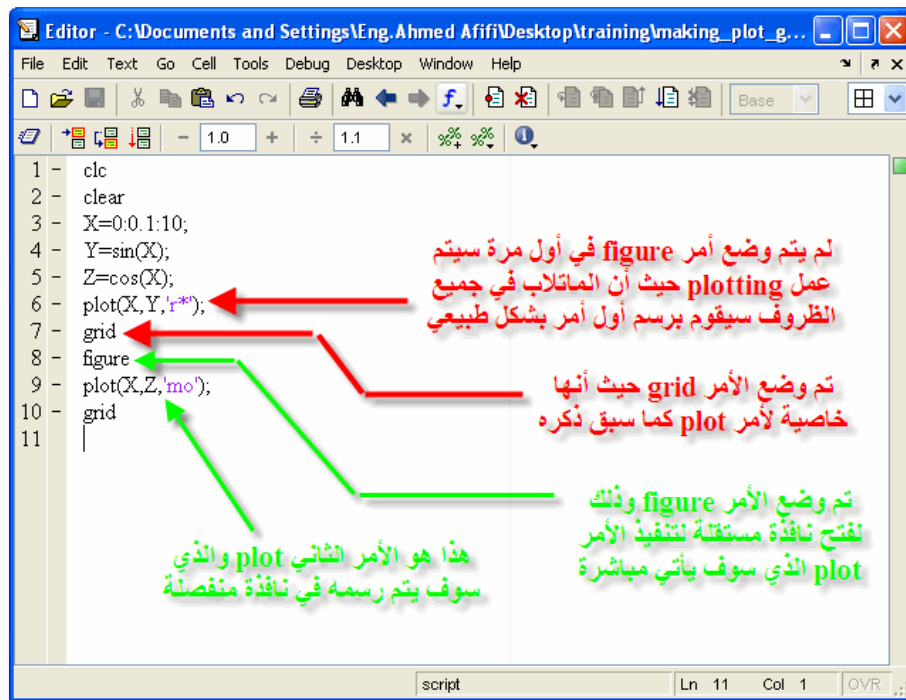
```

## وضع الرسومات في نوافذ منفصلة

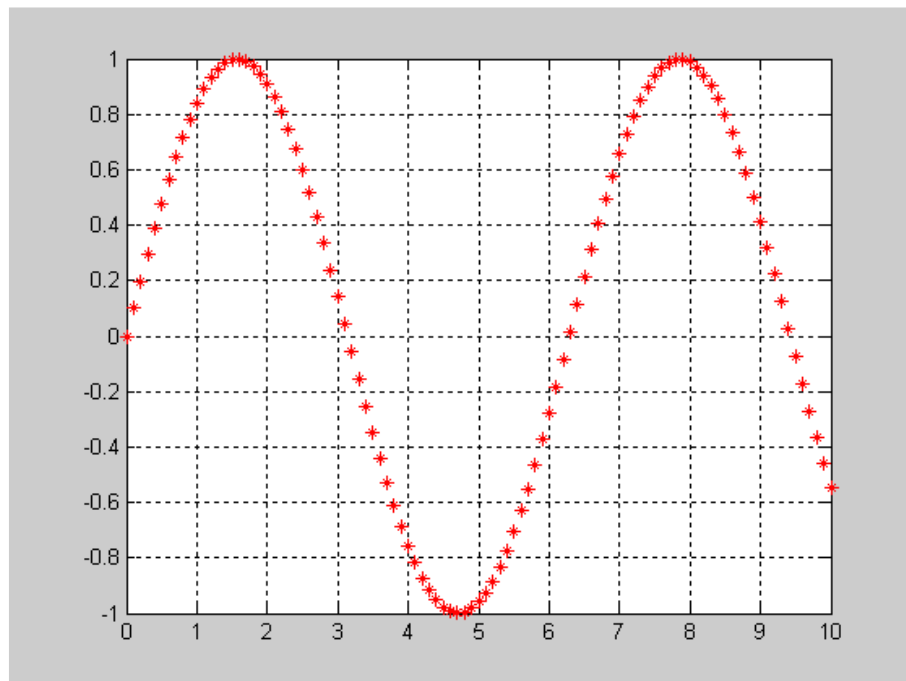
سنقوم الآن بدلاً من وضع الرسومات في نفس النافذة سنقوم بوضعها في نوافذ مختلفة وعليه نحتاج إلى الأمر `figure` والذي يقوم بفتح نافذة فارغة إذا تم وضعه منفصلاً، جرب ذلك في نافذة الأوامر ستلاحظ ان الماتلاب قام بإظهار نافذة رمادية اللون فارغة شاهد الصورة التالية

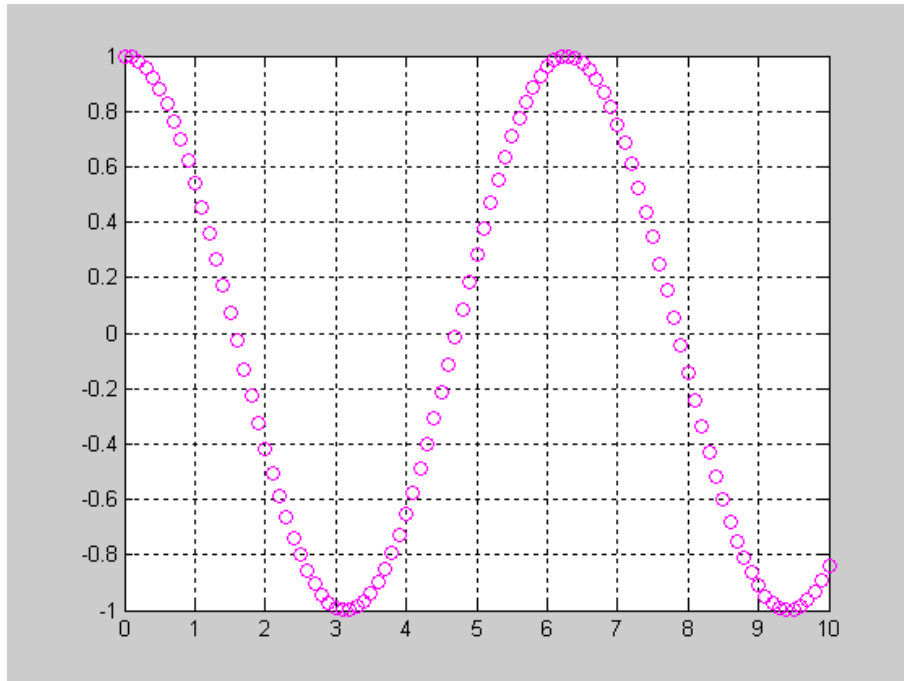


حيث وجود تلك النافذة يعني انه سيتم تنفيذ أمر الرسم `plot` الذي بعد أمر `figure` علماً أنه بعد كل أمر `figure` يتم وضع الخصائص التي تختص بهذه الرسمة مثل أمر `grid` الذي سبق شرحه. وهذا مثال بسيط على ذلك



وستحصل على نافذتين بهما كلتا الرسمتين





والآن قم بتشغيل البرنامج مرة أخرى، ستلاحظ أن عدد النوافذ قد زاد نافذة واحدة، فكيف حدث هذا؟ يقوم الماتلاب برسم أول دالة على النافذة الأخيرة التي تم رسم الدالة الثانية بها، ثم يقوم برسم الدالة الثانية في نافذة جديدة بسبب وجود الأمر figure ولحل هذه المشكلة قم باستخدام الأمر close all بعد الأمر clear بحيث يتم إغلاق أي نوافذ كانت مفتوحة قبل ذلك عند تشغيل البرنامج آل مرة وبالتالي سيكون هنالك ثلاثة أوامر لابد من استخدامها في كل مرة يتم عمل أي برنامج وهم

clc

clear

close all

وهذا هو المثال الذي تم عمله منذ قليل بعد التعديل

```

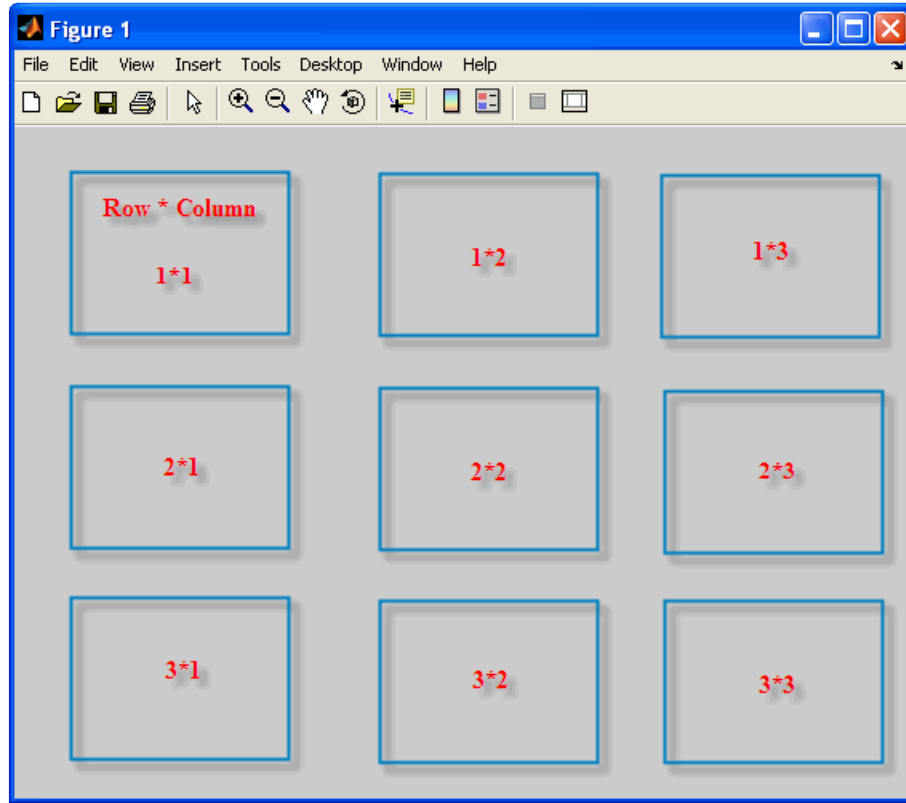
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Base
1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - Z=cos(X);
7 - plot(X,Y,'r*');
8 - grid
9 - figure
10 - plot(X,Z,'mo');
11 - grid
12

```

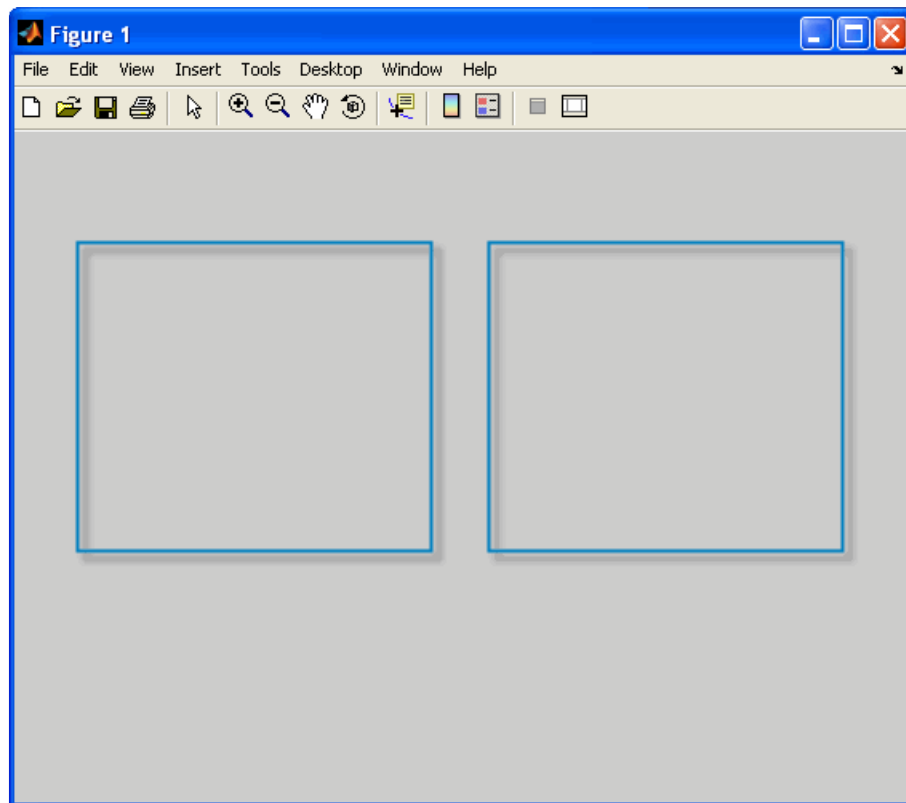
### إنشاء رسومات منفصلة في نافذة واحدة

بإمكاننا أن نقوم بعمل أكثر من رسمة على نفس النافذة، ويمكن أيضاً أن نقوم بوضع عدة رسومات منفصلة في نافذة واحدة، يمكن ذلك باستخدام الأمر subplot قبل كل أمر plot.

يعمل الأمر subplot من خلال تحديد عدد الرسومات التي ستقوم بإظهارها، حيث يقوم الأمر subplot على وضع الصور وكأنها مصفوفة أو متجه، ويجب عند استخدام الأمر معرفة عدد الرسومات التي ستظهرها وكيفية وضعها، ويفضل استخدام الأشكال التالية لتحديد الأماكن التي ستقوم بوضع الرسومات بها.



سنأخذ مثلاً، لنقول أن لدينا معادلات يجب رسمهما، وسنقوم بوضعهما بجوار بعضهما أما في الشكل التالي



وبالتالي الرسمتان سيكون وكأنهما متجه عدد صفوفها 1 وعدد الأعمدة 2 والرسم الأولى تأخذ الخانة الأولى، والرسم الثانية ستأخذ الخانة الثانية. هذا ما يجب تحديده بالتفصيل عند استخدام الأمر subplot ثم نستخدم الصورة العامة لأمر subplot والتي تكون كالتالي.

plot(number of rows, number of columns, the number of the matrix which occupy the figure)

ولرسم الشكل الأول لابد من كتابة الأمر في الصورة التالية

subplot(1,2,1)

رقم الخانة التي تشغلها

عدد الأعمدة

عدد الصفوف

ولرسم الشكل الثاني لابد من كتابة الأمر في الصورة التالية

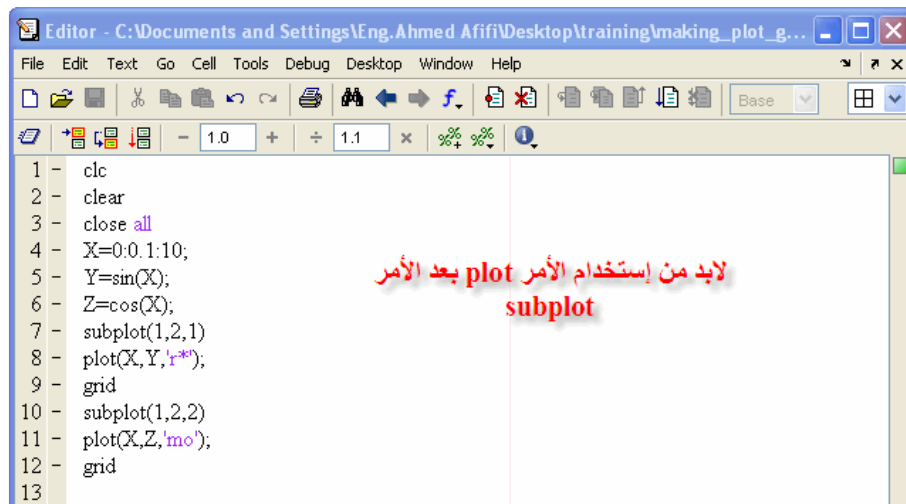
subplot(1,2,2)

رقم الخانة التي يشغلها الشكل

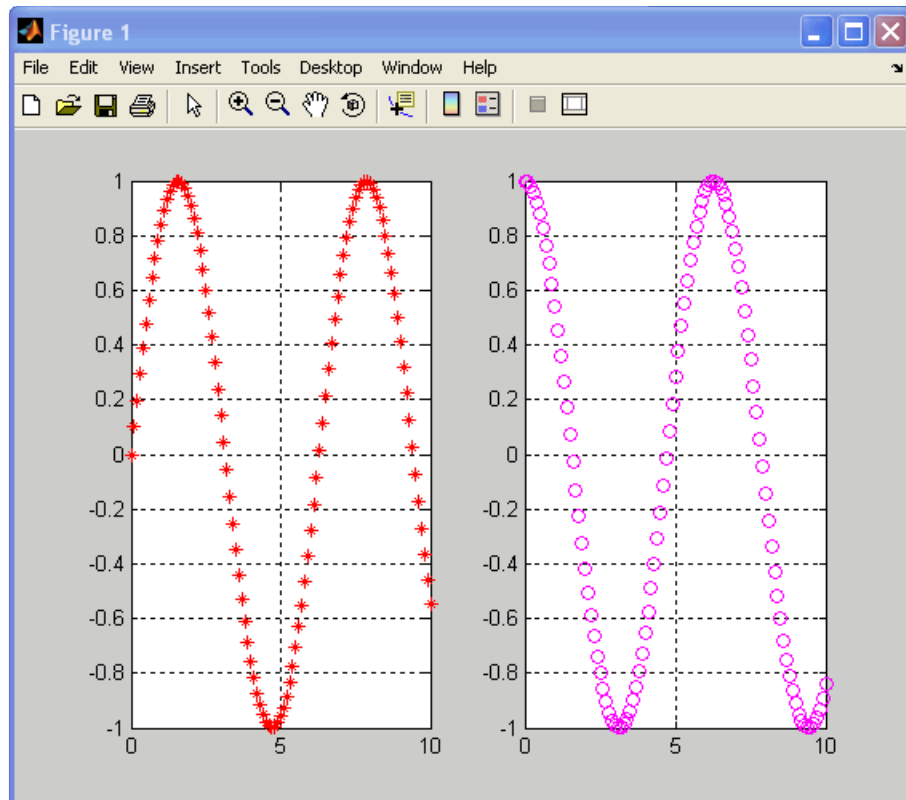
عدد الأعمدة

عدد الصفوف

والآن سنقوم بوضع البرنامج كاملاً ليكون المعنى قد وضح تماماً



وستكون الرسمة النهائية كما في الشكل التالي



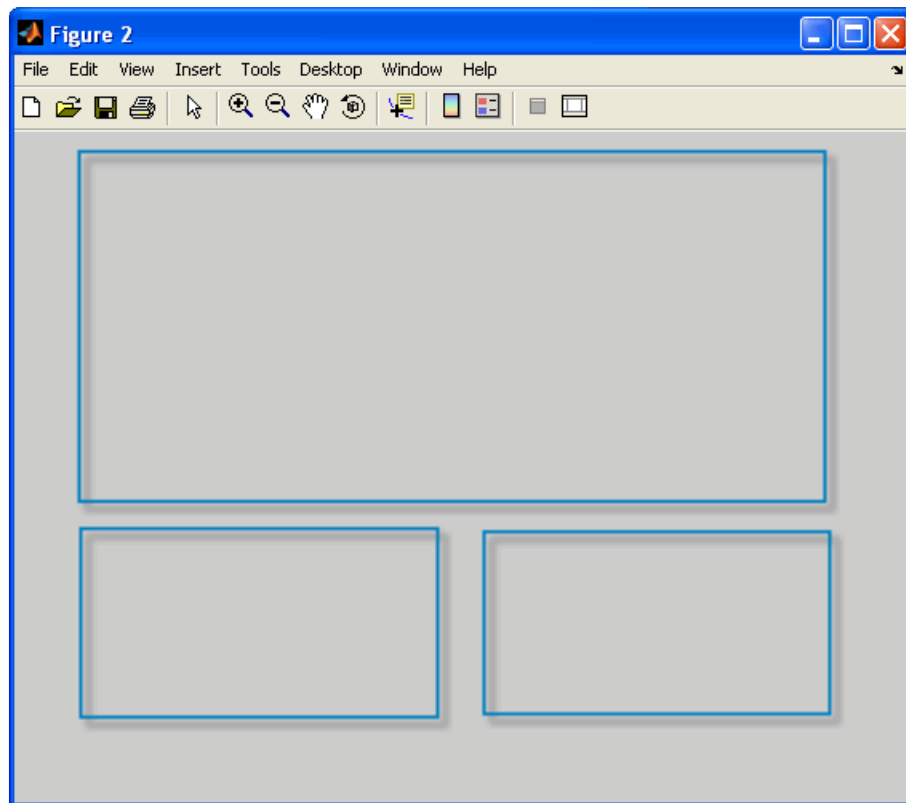
ملاحظة إذا كانت الرسمة تشغل أثر من خانة يتم استخدام الأقواس المربعة، وتأخذ الشكل التالي

[أرقام جميع الخانات التي تشغلها الرسمة]

وسنقوم بإعطاء مثال

نريد أن يكون الشكل الخارج على شكل الصورة التالية





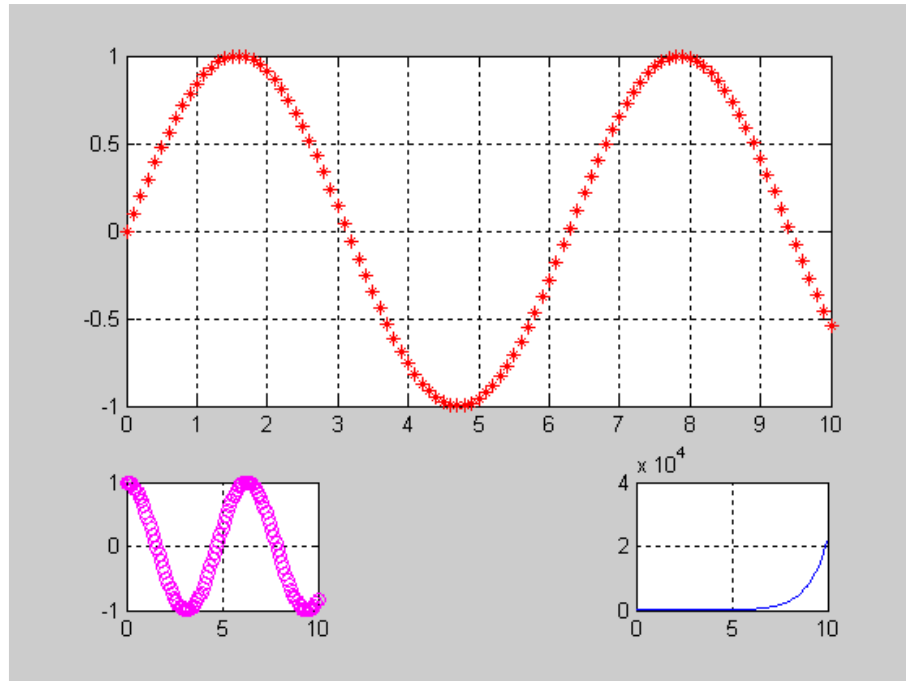
فإن عدد الصفوف 3 وعدد الأعمدة 3 وأرقام الخانات التي تشغلها الرسوم الأولى 1، 2، 3، 4، 5 و6 على التوالي وأرقام الخانات التي تشغلها الرسوم الثانية 7 وأرقام الخانات التي تشغلها الرسوم الثالثة هي 9. البرنامج يكون بالشكل التالي.

```

1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - Z=cos(X);
7 - V=exp(X);
8 - subplot(3,3,[1 2 3 4 5 6])
9 - plot(X,Y,'r*');
10 - grid
11 - subplot(3,3,7)
12 - plot(X,Z,'mo');
13 - grid
14 - subplot(3,3,9)
15 - plot(X,V);
16 - grid
17 -

```

وستكون النتيجة كالتالي



### تسمية المحاور

سنقوم الآن بتنفيذ الجزء قبل الأخير وهو تسمية المحاور، فمثلاً إذا أردنا أن نقوم بتسمية محور السينات X-Axis نقوم باستخدام الأمر xlabel وإذا أردنا أن نقوم بتسمية محور الصادات نقوم باستخدام الأمر ylabel حيث يأخذ كلا الأمرين صورة واحدة وهي كالتالي.

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

كما ترى لابد من أن يكون اسم المحور  
بين فاصلتين كما هو واضح بالشكل

نفس الشيء يتم تطبيقه على محور الصادات ylabel.

### وضع عنوان في أعلى الرسمة

يمكن وضع عنوان أعلى كل رسمة وذلك من خلال الأمر title حيث يكون هذا الأمر بالشكل التالي.

`Title('The title of the graph')`

لابد من وضع العنوان بين فاصلتين كما هو  
موضح بالرسم

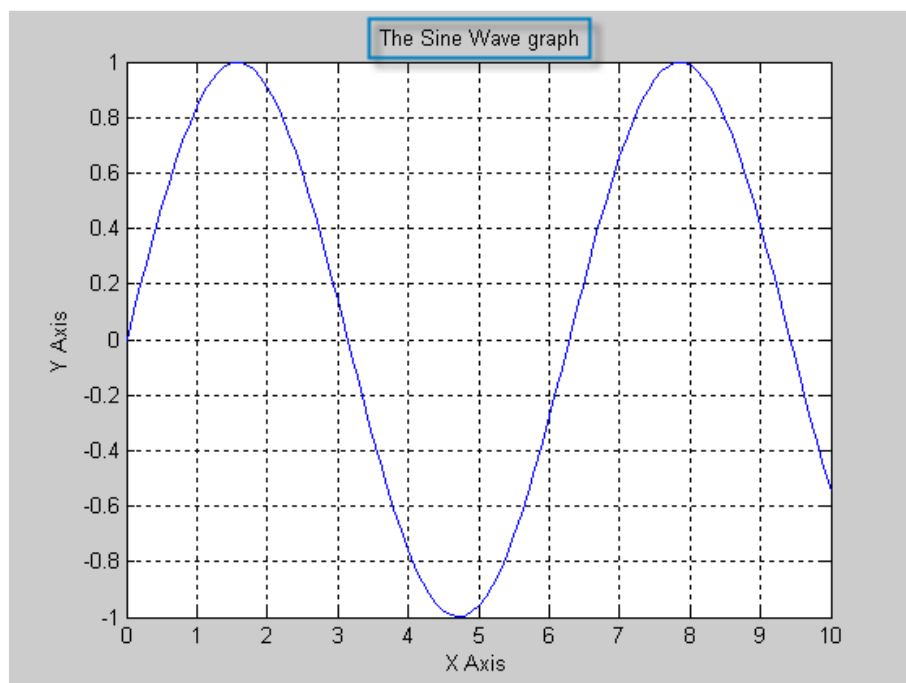
ولنفقم الآن بعمل مثال تطبيقي

```

1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - plot(X,Y,'b');
7 - xlabel(' X Axis ');
8 - ylabel(' Y Axis ');
9 - title(' The Sine Wave graph ');
10 - grid
11

```

وبالتالي نحصل على الصورة التالية



### وضع نص على نقطة أو أكثر داخل الرسم

يمكن إضافة نص على نقطة أو أكثر على الرسم، وذلك باستخدام الأمر text ويأخذ الصورة التالية

Text (position of the point at X-Axis, position of Y-Axis, ' The text on that point ')

يجب وضع النص بين فاصلتين

وسنأخذ مثلاً بسيطاً في كيفية إيجاد الرقم الأكبر، ثم وضع دائرة حمراء حول النقطة العظمى ووضع كلمة maximum point.

ولكن يجب أولاً شرح الأمر `find`. هذا الأمر يقوم بإيجاد مكان العنصر داخل المتجه بمجرد تحديد خصائص هذا العنصر، فمثلاً سنقوم بعمل دالة وسنبحث على العنصر الأكبر بينها كما في البرنامج التالي.

```

1  clc
2  clear
3  close all
4  x=linspace(0,10,100);
5  y=sin(x).*exp(-0.3*x);
6  ymax=max(y);
7  ind=find(y==ymax);
8

```

١ - هنا نقوم بإيجاد العنصر الأكبر داخل دالة

٢ - هنا نقوم بتحديد الرقم الأكبر داخل المتجه

٣ - لا بد من كتابة `==` حيث إنها تعني أننا نبحث على هذا العنصر بالتحديد دون أي عناصر أخرى، وهذا بما يسمى Condition أو

وعند تشغيل البرنامج، نجد القيم كالتالي

Name	Value	Class
ind	14	double
x	<1x100 double>	double
y	<1x100 double>	double
ymax	0.6521	double

وهذا هو مكان القيمة العظمى داخل المتجه

هذه القيمة العظمى

وبالتالي إذا أردنا الحصول على قيمة `x` عند القيمة العظمى `y` سنقوم بعمل التالي

```

1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*exp(-0.3*x);
6 - ymax=max(y);
7 - ind=find(y==ymax);
8 - xmax=x(ind);
9 -

```

قمنا بإيجاد قيمة X عند القيمة العظمى Y عن طريق وضع الصورة التالية  
 $X(ind)$   
 أي إيجاد X عند الخانة التي تعطي Y

وكما تلاحظ فإن قيمة X والتي تعطي القيمة العظمى Y تظهر في workspace.

Name	Value	Class
ind	14	double
x	<1x100 double>	double
xmax	1.3131	double
y	<1x100 double>	double
ymax	0.6521	double

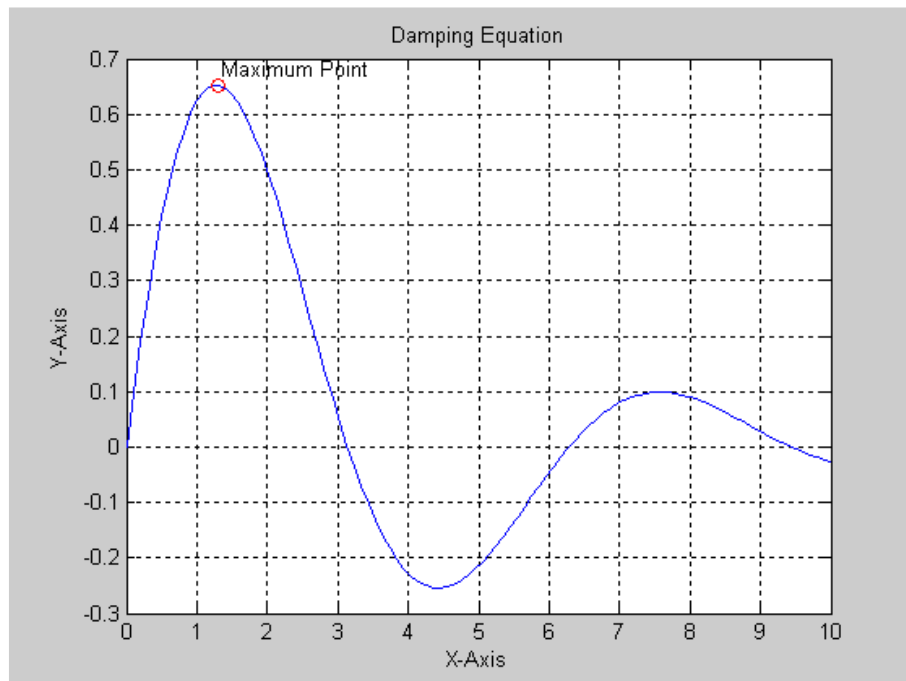
والآن سنقوم بتطبيق المثال ووضع كلمة النقطة العظمى عليها

```

1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*exp(-0.3*x);
6 - ymax=max(y);
7 - ind=find(y==ymax);
8 - xmax=x(ind);
9 - plot(x,y,xmax,ymax,'ro');
10 - title('Damping Equation');
11 - xlabel('X-Axis');
12 - ylabel('Y-Axis');
13 - grid
14 - text(xmax+0.03,ymax+0.03,'Maximum Point');

```

وبالتالي ستظهر الرسمة كما يلي



الأمر legend

يستخدم هذا الأمر في وضع دليل على صفحة الرسم ليبين ماذا يعني كل لون على الرسم، فمثلاً سنقوم بوضع الأمر legend في المثال التالي علماً أن هذا الأمر لابد من أن يأخذ الصورة

Legend ('the color reference')

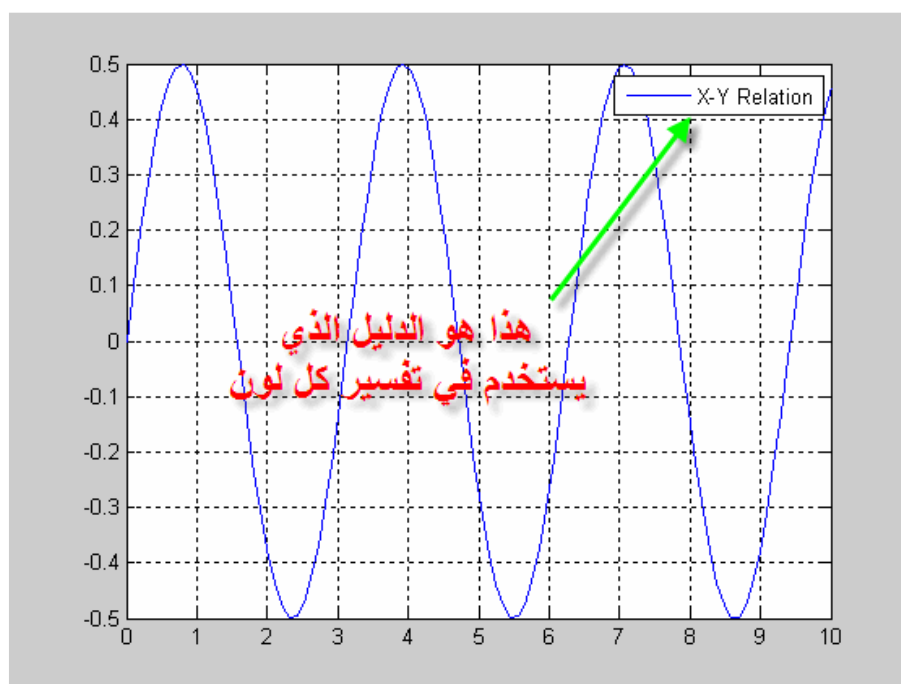
ويمكن كتابة البرنامج التالي على الماتلاب

```

C:\Program Files\MATLAB\R2006a\work\testi...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B...
1.0 1.1 x % % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*cos(x);
6 - title('Making legend Command');
7 - xlabel('X-Axis');
8 - ylabel('Y-Axis');
9 - plot(x,y);
10 - legend('X-Y Relation');
11 - grid
script Ln 11 Col 5 OVR

```

وبالتالي ستجد الناتج كالتالي



كما ترى فإن الأمر legend يعتمد على عدد العلاقات المرسومة داخل الرسم، فمثلاً المثال الذي سبق أخذه كان يستخدم في رسم علاقة ثم إيجاد النقطة العظمى أي أن عدد العلاقات المرسومة اثنان، وبالتالي تتم برمجته بالشكل التالي

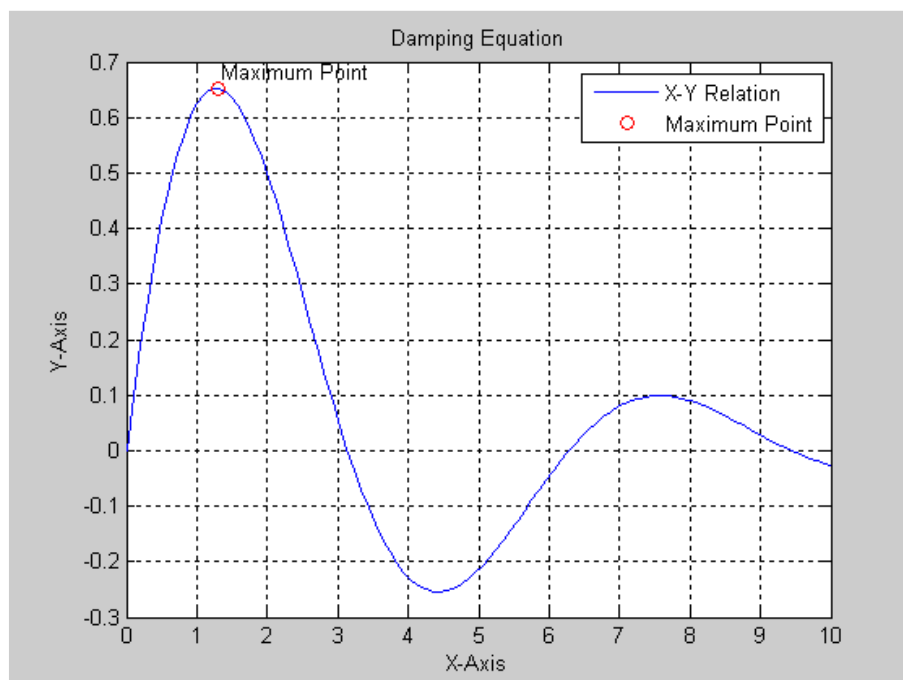
```

C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\Tr...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B... >>
1.0 1.1 x % % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*exp(-0.3*x);
6 - ymax=max(y);
7 - ind=find(y==ymax);
8 - xmax=x(ind);
9 - plot(x,y,xmax,ymax,'ro');
10 - title('Damping Equation');
11 - xlabel('X-Axis');
12 - ylabel('Y-Axis');
13 - grid
14 - text(xmax+0.03,ymax+0.03,'Maximum Point');
15 - legend('X-Y Relation','Maximum Point');
16
script Ln 16 Col 1 OVR

```

للعلاقة الأولى  
فاصل  
للعلاقة الثانية

وبالتالي تكون الرسمة كالتالي



ويجب مراعاة أن يتم استخدام الأمر legend بعد الأمر plot وليس العكس.



## فتح نافذة جديدة وتحديد دقتها

يعطي الماتلاب القدرة على فتح نافذة جديدة وتحديد القيم العظمى والصغرى لمحور السينات وكذلك بالنسبة لمحور الصادات، وذلك باستخدام الأمر `axis`، والذي يأخذ الصورة التالية في كتابته

`axis([minimum value of X, maximum value of X, minimum value of Y, maximum value of Y])`

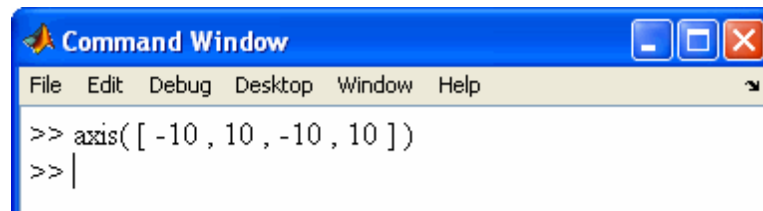
## مثال تطبيقي

قم بفتح نافذة للرسم بحيث تكون بالمواصفات التالية

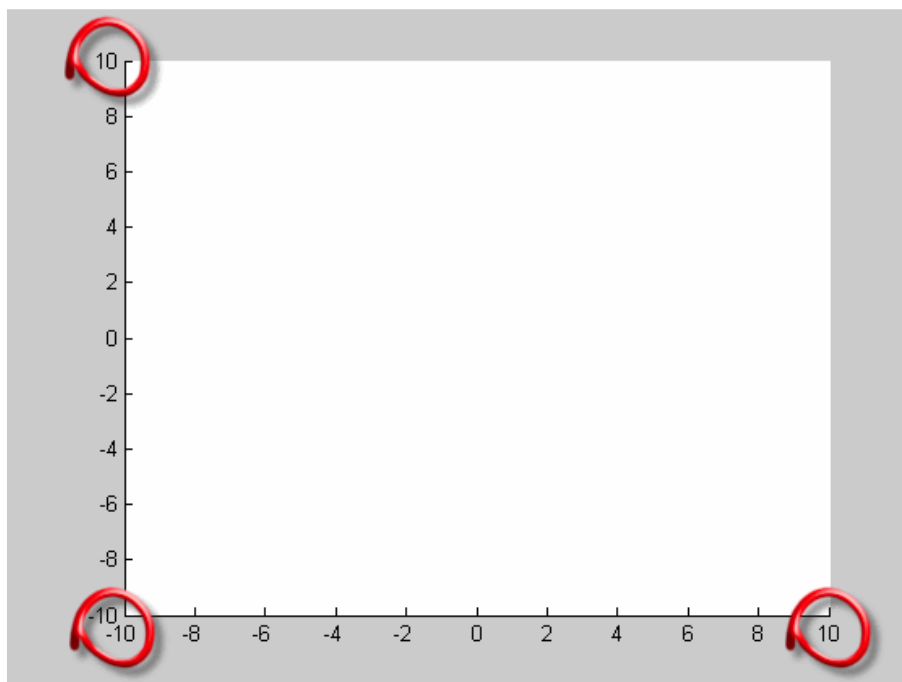
١. أقل قيمة لمحور السينات هي -10
٢. أعلى قيمة لمحور السينات هي 10
٣. أقل قيمة لمحور الصادات هي -10
٤. أعلى قيمة لمحور الصادات هي 10

خطوات الحل

في نافذة الأوامر قم بإدخال التالي



وستظهر لك النافذة التالية



## كيفية إدخال النقاط من خلال الماوس

تعلمنا أنه يمكننا إدخال القيم باستخدام المتجهات أو المصفوفات، ولكن يوفر الماتلاب قدرة في إدخال النقاط من خلال الرسم باستخدام الماوس، ونظراً لأننا نقوم باختيار النقاط من على الرسم فهذا يعني أن النقاط التي يتم اختيارها يتم تمثيلها في قيمة في محور السينات وقيمة في محور الصادات، ويتم وضع قيم محاور السينات والصادات في صورة متجه.

يستخدم الأمر `ginput` في عملية إدخال النقاط باستخدام الماوس، ويتم كتابة ذلك الأمر في الصورة التالية.

عدد مرات الإدخال

`[x y]=ginput(number of entry)`

كل نقطة يتم اختيارها لها إحداثيان  
X & Y

أما إذا أردنا إدخال عدد لا نهائي من النقاط يمكن ذلك بعدم ذكر عدد نقاط الإدخال، كما في الشكل التالي.

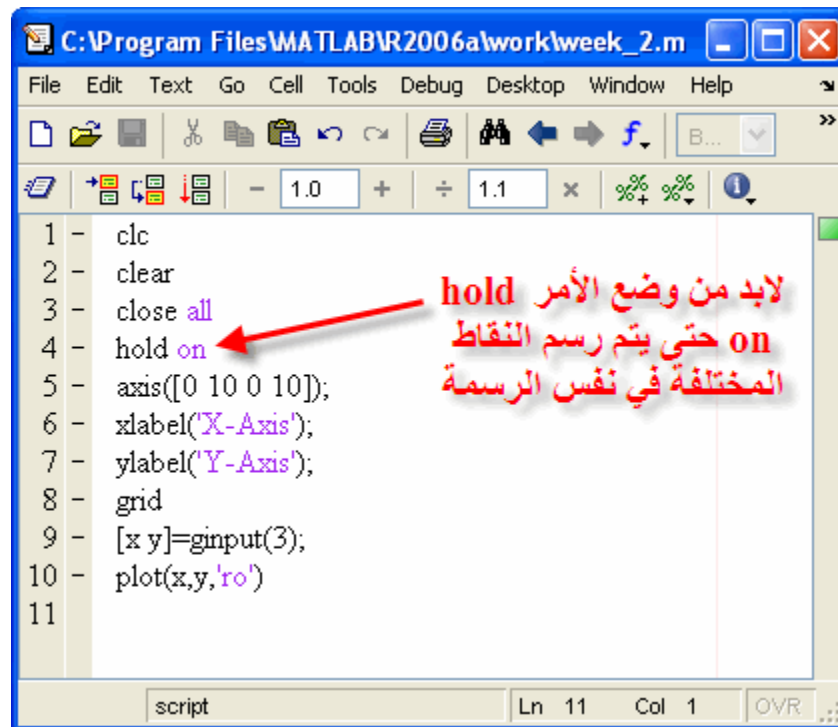
`[x y]=ginput()`

وذلك لإدخال عدد لا نهائي من النقاط

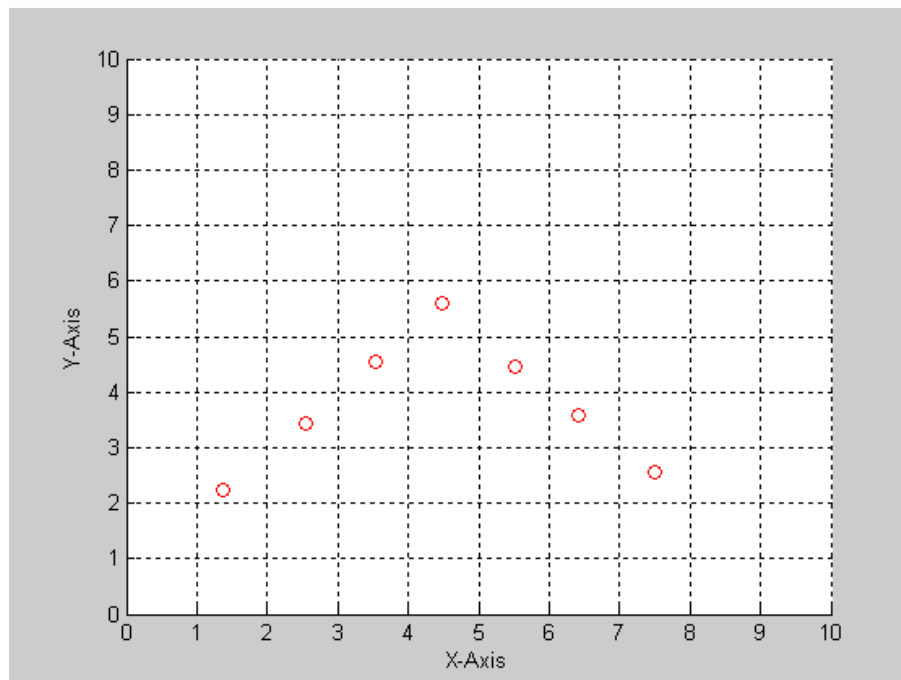
وبعد الإنهاء من إدخال النقاط كل ما عليك هو الضغط على مفتاح `Enter` في لوحة المفاتيح.

مثال تطبيقي

سنقوم بفتح نافذة للرسم بها شبكة، وأقل قيمة لمحور السينات هي صفر وأكبر قيمة لمحور السينات هي 10 وكذلك بالنسبة لمحور الصادات، ثم إدخال عدد كبير من النقاط على الرسم باستخدام الأمر `ginput`، وهذه النقاط يتم طباعتها على شكل دوائر حمراء. ويتم كتابة الأوامر بالشكل التالي.



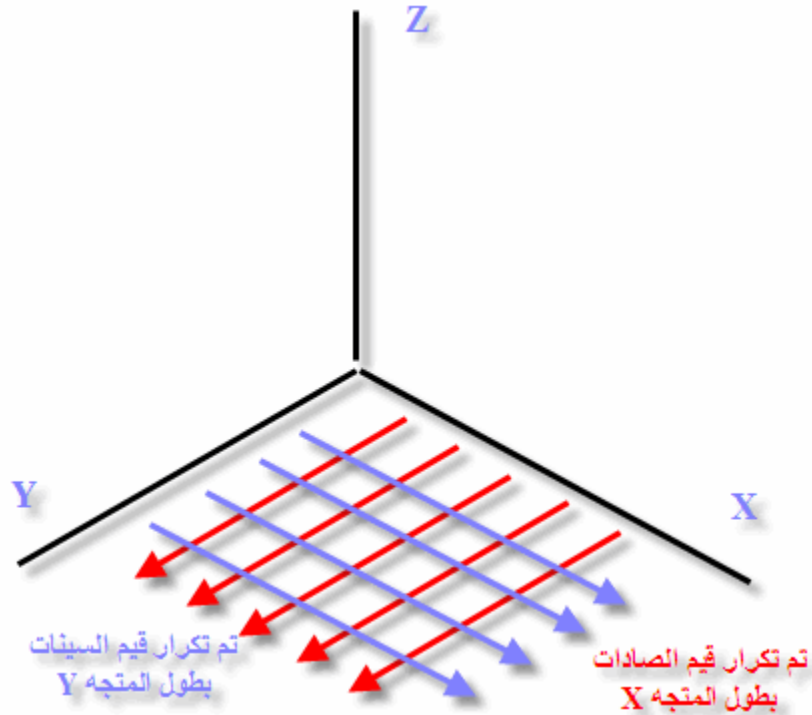
وستظهر لك نافذة لإدخال النقاط، وبعد إتمام عملية الإدخال اضغط على Enter لإتمام الإدخال. وستظهر لك النافذة التالية.



### الرسم ثلاثي الأبعاد

كما تعلمنا أن الرسم ثلاثي الأبعاد يعتمد على ثلاثة محاور لرسمها، محور  $X$ ،  $Y$  و  $Z$  وأن كلاً من  $X$  و  $Y$  يمثلان المستوى الأفقي، وأن المحور  $Z$  يمثل الارتفاع، وتلك القيم هي قيم النقاط الموجودة على المحاور، ولكن حتى يتم رسم أي نقطة في المستوى الأفقي يجب أن نقوم تعريف ذلك للماتلاب وذلك باستخدام الأمر meshgrid حيث يقوم الماتلاب بإنتاج مصفوفة يتم تكرار قيم محور السينات  $X$ -Axis بنفس طول محور

الصادات Y-Axis، كما يقوم بتكرار قيم محور الصادات Y-Axis بنفس طول قيم السينات X-Axis، وبهذا تكون المصفوفة المتكونة هي المستوى الأفقي أما هو واضح بالرسم التالي.



علما أن الأمر meshgrid يأخذ الصورة التالية في كتابته

$[x \ y] = \text{meshgrid}(x, y)$

وبعد استخدام الأمر meshgrid يتم استخدام الأمر mesh والذي يستخدم أبديل الأمر plot ولكن في الرسم ثلاثي الأبعاد.

مثال تطبيقي

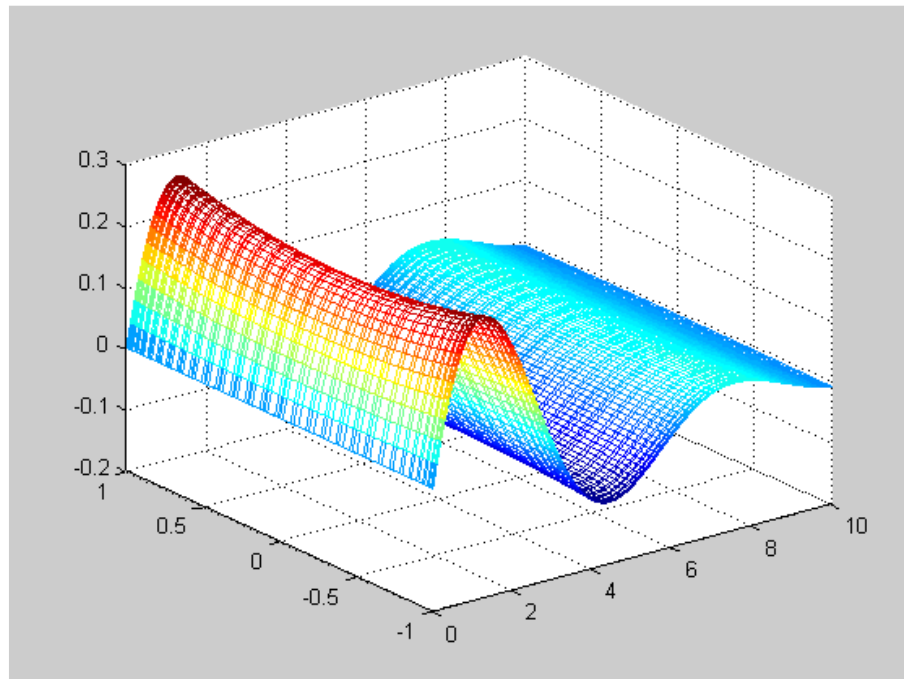
نقوم في هذا المثال بتعريف قيم محور السينات X-Axis وسنقوم بوضع المعادلة التي تصف محور الصادات وعلاقته بمحور السينات، أخيراً وليس آخراً نقوم بوضع العلاقة التي تربط بين محور السينات والصادات.

```

C:\Program Files\MATLAB\R2006a\work\week_2_...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B...
- 1.0 + ÷ 1.1 x % % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x);
6 - [x y]=meshgrid(x,y);
7 - z=sin(x).*exp(-0.3*x)./(cos(y)+2);
8 - mesh(x,y,z);
9
script Ln 4 Col 20 OVR

```

وبالتالي تكون الرسمة الناتجة كالتالي



أما ترى فإن الرسمة الناتجة عبارة عن شبكة تعتمد مجموعة النقاط لكلاً من  $x$  و  $y$  فإذا قمنا بزيادة عدد نقاط وبالتالي تزداد قيمة  $y$  كذلك

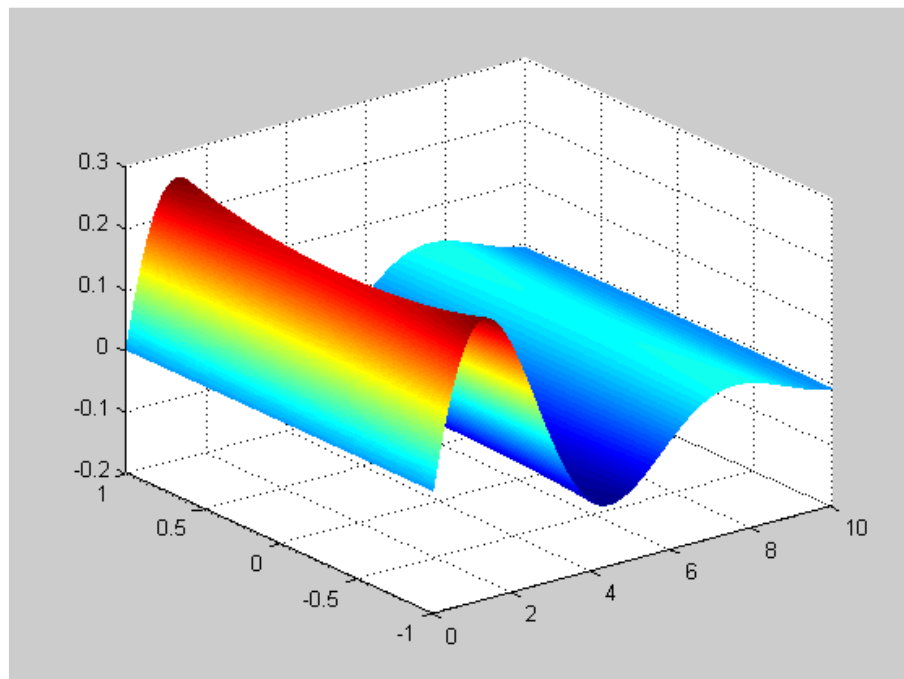
```

C:\Program Files\MATLAB\R2006a\work\week_2_...
File Edit Text Go Cell Tools Debug Desktop Window Help
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,1000);
5 - y=sin(x);
6 - [x y]=meshgrid(x,y);
7 - z=sin(x).*exp(-0.3*x)./(cos(y)+2);
8 - mesh(x,y,z);
9 -
script Ln 9 Col 1 OVR

```

كما ترى فإن عدد النقاط تم زيادتها من ١٠٠ إلى ١٠٠٠

وبالتالي فإن الرسمة الناتجة تكون كالتالي



أعتقد أنك تلاحظ الفرق الآن

ملاحظة ألما زادت عدد النقاط كلما زاد الوقت المستغرق لإظهار النتائج في الماتلاب