

6. العمليات المنطقية

توجد ثلاث عمليات رئيسية للمقارنة بين مصفوفتين أو للمتجهات أو مصفوفة مع قيمة عددية أو بين القيم العددية

الرمز	العملية	العلاقة
&	And	و
	Or	أو
~	not	النفي

```
>> a=[1 0 1];  
>> b=[0 1 0];
```

```
>> H= a&b
```

```
H=  
0 0 0
```

```
>> h= a|b
```

```
h=  
1 1 1
```

```
>> h=~a
```

```
h=  
0 1 0
```

```
>> H=~b
```

```
H=  
1 0 1
```

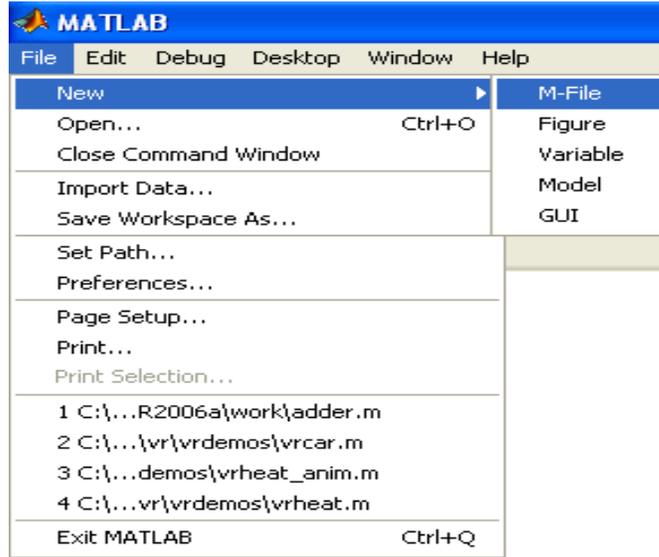
نفس العمليات يمكن تطبيقها على مصفوفات ذات بعدين

H.W

إذا كان $a=[1\ 0;\ 0\ 1]$ و $b=[0\ 0;\ 1\ 1]$ قم بتطبيق العمليات المنطقية الثلاثة بين المصفوفتين

7. ملفات M التخطيطية M-file

ان إدخال الأوامر إلى ماتلاب عبر النافذة command سريع وفعال من اجل المسائل البسيطة , لكن عند ازدياد عدد الأوامر او عندما نرغب بإعادة تنفيذ عدد من الأوامر مع تغيير قيمة متغير او اكثر فان ادخال الأوامر عبر نافذة الاوامر command window يعد امرا مرهقا. يؤمن ماتلاب حلا منطقيا لهذه المشكلة فهو يسمح بكتابة الاوامر او الخوارزمية في ملف نصي بسيط ليتم فيما بعد اخبار Matlab بان عليه فتح هذا الملف. ولفتح نافذة M-file نتبع التالي



تعتبر M-file هي وسيلة لإدخال الأوامر ولكن ليس من خلال نافذة الأوامر, فماذا قد يختلف في هذه الوسيلة الجديدة في إدخال الأوامر؟

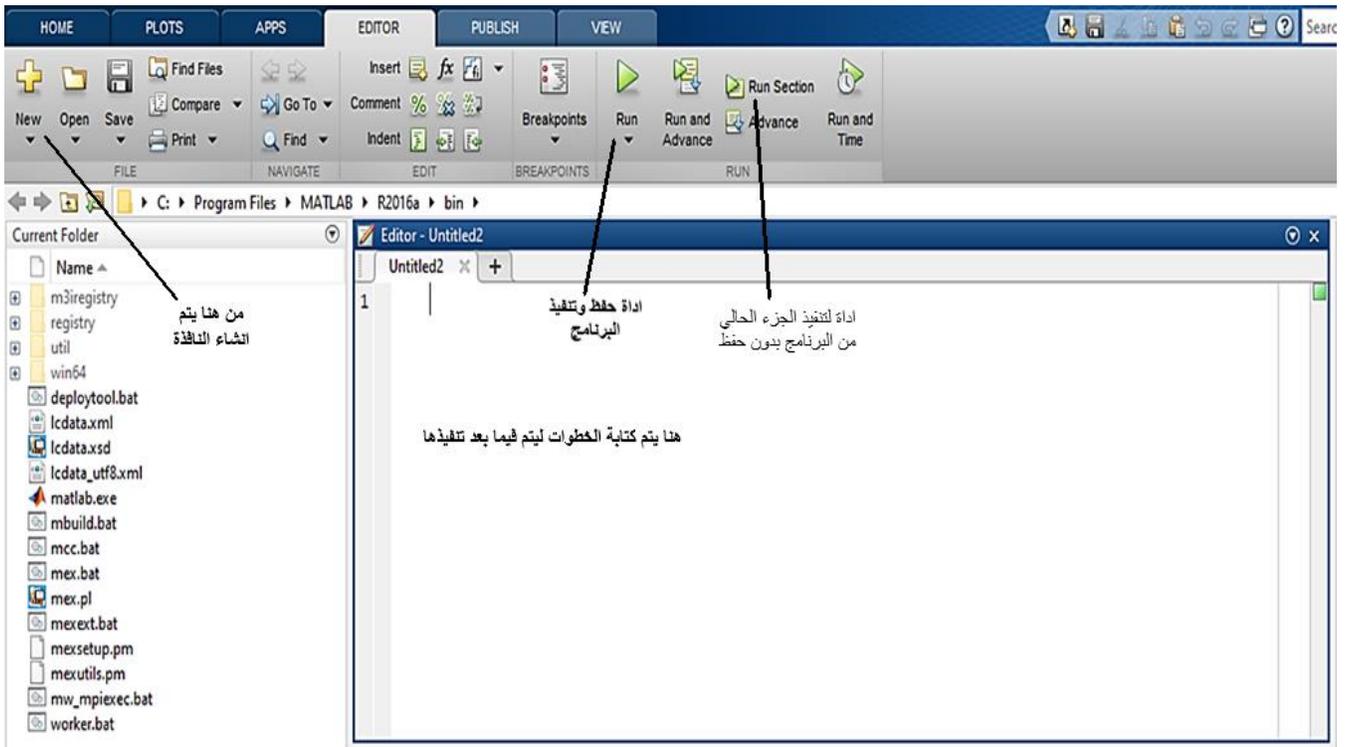
1 - في عملية إدخال الأوامر التي كنا نستخدمها, إذا أردنا تعديل عنصر أو أكثر كان يجب إعادة إدخال الأمر من جديد.

2 - إذا كتبنا برنامج كبير, وأردنا إعادة العملية مرة أخرى يجب إدخال جميع الأوامر من جديد وبنفس الترتيب.

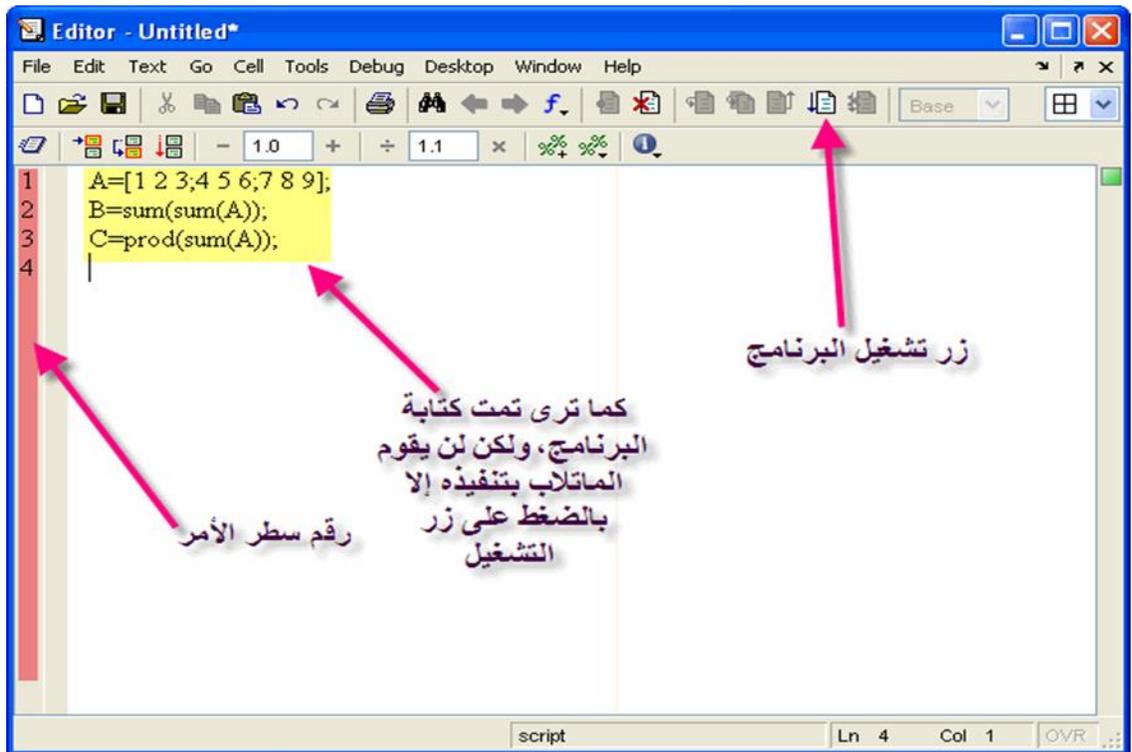
3 - عمل عملية تصحيح للأخطاء Debugging يكون صعب بدون وجود نافذة M-file وهذا بالطبع يستغرق وقتاً كبيراً هذا بالإضافة إلى الملل الذي يحدث للمستخدم, وطبعاً حلاً لهذه المشكلة, تم عمل ما يسمى M-File والتي تعطي القدرة على كتابة البرنامج كاملاً أولاً بدون تشغيل, وبعد الانتهاء منه يتم تشغيله.

هذه الخاصية تعطي القدرة على تعديل القيم دون الحاجة إلى كتابتها مرة أخرى, أو إعادة إدخال الأوامر التي تعتمد على هذا.

سنقوم الآن بالتعرف على نافذة M-File, أنظر الصورة التالية :



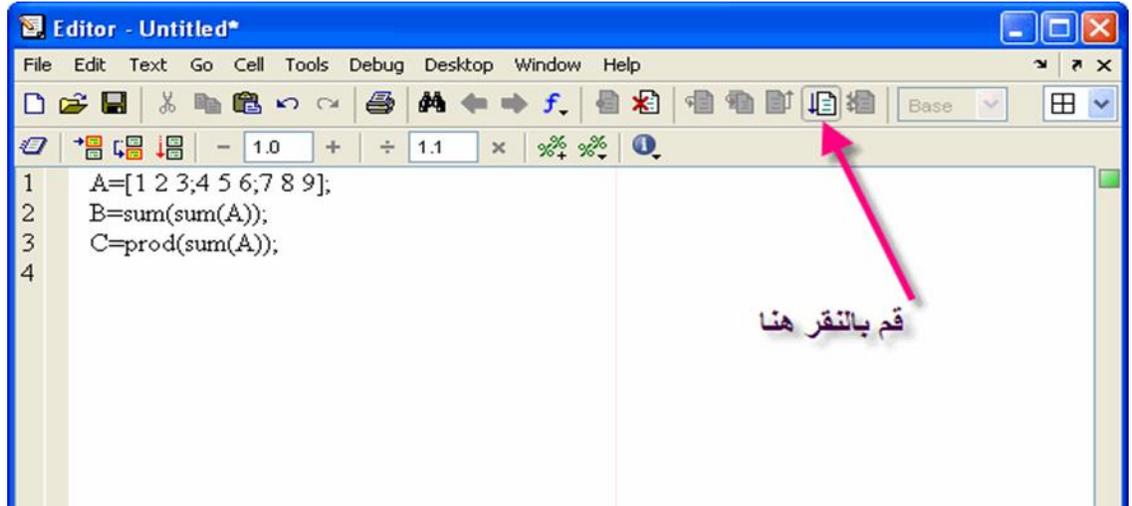
شكل نافذة m-file (او تدعى نافذة محرر النصوص) في الاصدارات القديمة لماتلاب



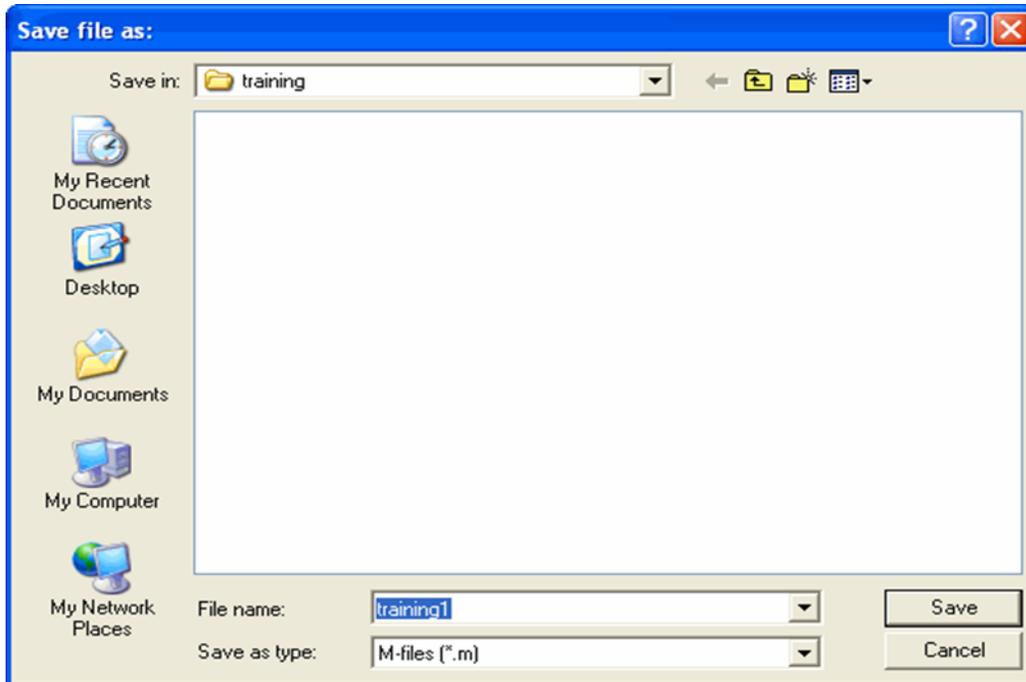
- ولكن عند الضغط على زر التشغيل، سيطلبك الماتلاب بحفظ البرنامج، ولكن يشترط الآتي عند حفظ البرنامج
- 1 - أن لا يبدأ بأرقام
 - 2 - أن لا يكون أمراً معرّفاً في الماتلاب

- 3 - أن لا يحتوي الاسم على مسافات فاصلة
4 - أن لا تحتوي على رموز خاصة مثل * , & , - , +

يجب مراعاة تلك الشروط و إلا لن يقوم الماتلاب بتنفيذ البرنامج فالنقم بتنفيذ المثال المكتوب الآن في النافذة السابقة
1 - يتم الضغط على زر التشغيل كما هو واضح في الصورة التالية



2 - سيطلبنا الماتلاب بحفظ البرنامج أولاً, ولنسميه training1



3 - ستظهر القيم في كلاً من Command Window and Workspace

The image displays the MATLAB software interface. At the top, the title bar reads "MATLAB" and "EN Eng". The menu bar includes "File", "Edit", "View", "Graphics", "Debug", "Desktop", "Window", and "Help". The current directory is "C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training".

The **Workspace** window shows three variables:

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

The **Editor** window shows the following code:

```

1 - A=[1 2 3;4 5 6;7 8 9]
2 - B=sum(sum(A))
3 - C=prod(sum(A))
4

```

The **Command History** window shows the following commands:

```

B=sum(diag(A))
clc
A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]
B=prod(diag(A))
clc
clear
clc
help magic
magic(3)
magic(9)
clc
A=magic(3)
B=magic(9)
clc

```

The **Command Window** shows the output:

```

A =
    1     2     3
    4     5     6
    7     8     9

B =
    45

C =
    3240

```

4 - لنعود إلى M-File ونقوم بتغيير بعض القيم للمصفوفة, كما في الشكل التالي

```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Base [Grid]
[Icons] - 1.0 + ÷ 1.1 x % % !
1 - A=[11 22 33;41 51 61;78 88 98]
2 - B=sum(sum(A))
3 - C=prod(sum(A))
4 - |
script Ln 4 Col 1 OVR
```

5 - سنقوم الآن بتشغيل البرنامج, وسيقوم الماتلاب الآن بالحفظ تلقائياً دون الحاجة لإعادة التسمية, ثم شاهد نافذة الأوامر Command Window

```
Command Window

1  2  3
4  5  6
7  8  9

B =
45
هذه قيم البرامج التي قد حصلنا عليها
منذ قليل

C =
3240

A =
11  22  33
41  51  61
78  88  98

B =
483
وهذه قيم البرنامج بعد عمل
التعديلات عليه

C =
4018560
```

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 يكون مبدأ في جميع البرامج التي نقوم بعملها لابد من أن تبدأ بهذا الأمر، ودعونا نقوم بمثال يوضح لنا ذلك

سنقوم الآن بكتابة مثال آخر في الـ M-File

```
1 - clc
2 - A=[1 2 3;4 5 6;7 8 9]
3 - B=sum(sum(A))
4 - C=prod(sum(A))
5
```

Command Window

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

B =
45

C =
3240
```

سنقوم الآن بتعديل المثال, وحتى نتأكد أن أمر `clc` يعمل, ستختفي القيم من `Command Window` وتظهر القيم الجديدة
وبهذا نتأكد من أن الأمر `CLC` يعمل بكفاءة ولكن دعونا نشاهد نافذة `Workspace` والتي تحتوى على قيم `A,B,C`

Workspace

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

Editor - C:\Documents and Settings\Eng.Ahmed Affif\Desktop\training\training

```
1 - clc
2 - A=[1 2 3;4 5 6;7 8 9];
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 إختفت وظهرت القيم الجديدة

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

Workspace

قيم البرنامج الأول قبل تغيير الرموز لاتزال 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 بإذن الله. وسنقوم الآن بتنفيذ نفس البرنامج ولكن بعد وضع الأمر `clear`, وستلاحظ الفرق الشاسع في الماتلاب الآن

كما ترى إختفت الرموز القديمة باستخدام الأمر `clear`

ولأيجاد محدد المصفوفة، شرط ان تكون المصفوفة مربعة، نستخدم الامر `det(name of matrix)`

مثلا لأيجاد محدد المصفوفة $A=[1 \ 2; \ 2 \ 0]$ نكتب الامر `det(A)`

س1// اكتب برنامج لأيجاد مجموع عناصر متجه ما ؟

س2// اكتب برنامج يقوم بتقريب عناصر المصفوفة نحو الصفر ؟ ثم جد محدد تلك المصفوفة ؟

س3// اكتب برنامج يقوم بتقريب عناصر مصفوفة ما باتجاه العدد الصحيح

