81 100 121 144

ولقد وجدنا هنا مربع كل عنصر من عناصر المصفوفة g.

المصفوفات القياسية

يمكّنك برنامج MATLAB من إنشاء مصفوفات قياسية، وذلك لتمتع تلك المصفوفات بخواص وميزات خاصة، وتتضمن أيضاً المصفوفات التي جميع عناصرها صفرية أو مساوية للواحد، ومصفوفات الأعداد العشوائية والمصفوفات القطرية والمصفوفات التي عناصرها أعداد ثابتة.

```
>> ones (3)          (مصفوفة واحديه)
ans =
    1   1   1
    1   1   1
    1   1   1
>> zeros (2, 4)       (مصفوفة صفرية)
        ↑     ↖
      الأسطر   الأعمدة
ans =
    0   0   0   0
    0   0   0   0
>> size (g)           (تحديد أبعاد مصفوفة)
ans =
     3   4
>> ones (size (g))
ans =
    1   1   1   1
    1   1   1   1
    1   1   1   1
```

ملاحظة: عندما يتبع اسم المصفوفة القياسية برقم مفرد مثل ones (n) أو zeros (n) فأن برنامج MATLAB ينشيء مصفوفات مربعة $n \times n$ تحتوي على أصفارا أو واحديه على الترتيب.

```
>> eye (4)            (مصفوفة الوحدة)
```

```
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
>> rand (3)          (مصفوفة عشوائية)

ans =
    0.9501    0.4860    0.4565
    0.2311    0.8913    0.0185
    0.6068    0.7621    0.8214
>> rand (1, 5)
ans =
    0.4447    0.6154    0.7919    0.9218    0.7382
```

مثال:

```
>> d = pi;
>> d * ones (3, 4)
ans =
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
>> d + zeros (3, 4)
ans =
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
>> repmat (d, 3, 4)        (تكرار القيمة d بالأبعاد 4×3)
```

```
ans =
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
```

ملاحظة: يمكن ان تكون d مصفوفة فتكون حينئذ تكرر مصفوفات وليس قيم.

التعامل مع المصفوفة

لقد امتلك برنامج MATLAB العديد من الطرق للتعامل مع المصفوفات، وكانت هذه الخاصية هي أهم مميزات البرنامج، فما إن تُحدّد المصفوفة حتى يزودك البرنامج بأقوى طرق الإدخال، التوسعة أو إعادة ترتيب بعض أجزاء المصفوفة عبر استعمال تعابير أو تعليمات محددة وممتعة، وتعتبر معرفة هذه التعليمات مفتاح الاستعمال الفعال لبرنامج MATLAB. ولشرح التعامل مع المصفوفات نأخذ الأمثلة التالية:

```
>> A = [1  2  3; 4  5  6; 7  8  9]
A =
    1   2   3
    4   5   6
    7   8   9
>> A (3, 3) = 0
A =
    1   2   3
    4   5   6
    7   8   0
```

جعل العنصر في الموقع (3, 3) صفراً.

```
>> A (2, 6) = 1
A =
    1   2   3   0   0   0
```

جعل العنصر في الموقع (2, 6) تكون 1 وبما ان المصفوفة A لا تمتلك ستة أعمدة، لذلك سيقوم البرنامج بتوسيعها حسب الضرورة ويضع بقي العناصر صفراً وتكون مستطيلة.

```
    4   5   6   0   0   1
    7   8   0   0   0   0
>> A (:, 4) = 4
A =
    1   2   3   4   0   0
    4   5   6   4   0   1
    7   8   0   4   0   0
```

جعل جميع عناصر العمود الرابع تكون 4

```
>> A (:, 4) = [4; 4; 4]
A =
    1   2   3   4   0   0
    4   5   6   4   0   1
    7   8   0   4   0   0
```

جعل جميع عناصر العمود الرابع تكون 4

```
>> A (:, 4) = [4  4  4]
    Error
```

بسبب عدم وجود فارزة منقوطة

<u>مثال:</u>

```
>> B = [7   8   9;  4   5   6; 1   2   3]
B =
    7   8   9
    4   5   6
    1   2   3
>> A = [1   2   3; 4   5   6; 7   8   9]
A =
    1   2   3
    4   5   6
    7   8   9
>> C = [A   B(:, [1  3])]
C =
```

حصلنا على المصفوفة C عبر توسيع المصفوفة A بإضافة العمودين الأول والثالث من المصفوفة B.

```
    1   2   3   7   9
    4   5   6   4   6
    7   8   9   1   3
>> B = A (1: 2, 2: 3)
B =
    2   3
    5   6
```

مثال: تشكيل المصفوفة B بجعل المصفوفة A كمصفوفة عمود واخذ أعمدتها عمود بعد عمود.

```
>> B = A (:)
B =
    1
    4
    7
    2
    5
    8
    3
    6
    9
```

مثال:

```
>> B = [1  2  3; 4  5  6; 7  8  9];
>> B = reshape (A, 1, 9)
B =
    1   4   7   2   5   8   3   6   9
>> B = reshape (A, [1  9]);
B =
   1   4   7   2   5   8   3   6   9
```

في المثال أعلاه إيعاز تحويل أبعاد المصفوفة الثنائية 3*3 إلى مصفوفة أحادية 9*1

مثال:

```
>> A = B
A =
   1   2   3
   4   5   6
   7   8   9
>> B (:, 2) = [ ]
B =
   1   3
   4   6
   7   9
```

تمت إعادة صياغة المصفوفة B عبر حذف كل اسطر العمود الثاني من المصفوفة B الأصلية، وعندما تضع أي عنصر مساوياً للمصفوفة الفارغة [ ]، فهذا يعني انك تريد حذفها من المصفوفة وتقليصها لتحافظ على العناصر المتبقية بعد الحذف.

مثال: إيجاد منقول (مدور) المصفوفة وإعادة تشكيلها بالتعليمة reshape.

```
>> C = B'
C =
   1   4   7
   3   6   9
>> reshape (B, 2, 3)
ans =
    1   7   6
    4   3   9
```

ملاحظة: تعمل تعليمة reshape عمود بعد عمود وللحصول على سطر بعد سطر نعمل المدور (transport).

مثال: هنا حذفنا السطر الثاني في المصفوفة C.

```
>> C (2, :) = [ ]
C =
   1   4   7
```

مثال: استبدلنا عناصر السطر الثاني من المصفوفة A بعناصر C.

```
>> A (2, :) = C
A =
     1   2   3
     1   4   7
     7   8   9
```

مثال:

```
>> x = -3: 3
x =
   -3   -2   -1   0   1   2   3
```

هناك المصفوفات المنطقية الناتجة عن العمليات المنطقية. كما يمكن أيضاً استخدام المصفوفات المنطقية إذا كان حجمها مساويا لحجم المصفوفات المعنونة، ويتم في هذه الحالة الإبقاء على العناصر ذات القيمة (1) أي true وهي العناصر المحققة للشرط بينما يتجاهل العناصر (0) أي false وهي العناصر غير المحققة الشرط. ولنأخذ المثال التالي:

```
>> abs (x) > 1
ans =
     1   1   0   0   0   1   1
>> y = x (abs (x) > 1)
```

هنا تم إنشاء المصفوفة y من تلك العناصر من المصفوفة x التي قيمتها اكبر من الواحد.

```
y =
    -3   -2   2   3
```

ويمكن العمل مع المصفوفات الثنائية المنطقية كما عملنا مع الأحادية المنطقية، كما في المثال التالي:

```
>> B = [5   -3; 2   -4]
B =
    5   -3
    2   -4
>> x = abs (B) > 2
```

```
x =

    1    1
    0    1
>>  y = B (x)
y =
    5
   -3
   -4
```

ترتيب المصفوفة

عندما تعطى متجه من البيانات فان أهم عملية يمكن إن نحتاجها وتود تطبيقها هي الترتيب، ويؤمن الايعاز sort عملية الترتيب في لغة MATLAB، كما هو واضح في المثال التالي:

```
>> x = randperm (8)                    (إيعاز ترتيب الأرقام بصورة عشوائية)
x =
    7    5    2    1    3    6    4    8
>> [y, indx] = sort (x)
```

الترتيب الجديد المواقع القديمة

```
y =
    1    2    3    4    5    6    7    8
indx =
    4    3    5    7    2    6    1    8
```

وعندما تكون المصفوفة ثنائية البعد فان عملية الترتيب تتم بشكل مختلف وكما يلي:

(عمود بعد عمود)

```
>> A = [randperm (6); randperm (6); randperm (6); randperm (6)]
```

```
A =
    1   2   5   6   4   3
    4   2   6   5   3   3
    2   3   6   1   4   5
    3   5   1   2   4   6
>> [As, idx] = sort (A)
As =
    1   2   1   1   3   1
    2   2   5   2   4   3
    3   3   6   5   4   5
    4   5   6   6   4   6
idx =
    1   1   4   3   2   2
    3   2   1   4   1   1
    4   3   2   2   3   3
    2   4   3   1   4   4
```

البحث عن مصفوفة جزئية

من المفيد في بعض الأحيان إن تعرف موقع أو دليل العناصر التي تحقق شرطا معينا، والموجودة ضمن مصفوفة معينة. يقوم برنامج MATLAB بتحقيق هذه الغاية عبر الايعاز find، والذي يعيد لك دليل أو موقع العنصر الذي تكون نتيجة تحقيقه لشرط ما true، واليك المثال التالي:

```
>> x = -3: 3
x =
    -3   -2   -1   0   1   2   3
>> k = find (abs (x) > 1)
 k =              (الموقع)
    1   2   6   7
```

```
>> y = x (k)
y =
    -3   -2   2   3
>> y = x (abs (x) > 1)
y =
    -3   -2   2   3
```

ويستطيع الايعاز find أن يعمل في المصفوفات الثنائية البعد أيضا (عمود بعد عمود)، فمثلا:

```
>> A = [1   2   3;   4   5   6;   7   8   9]
A =
    1   2   3
    4   5   6
    7   8   9
>> [i, j] = find (A > 6)
i =
    3
    3
    3
j =
    1
    2
    3
```

ملاحظة: الايعاز diag يوجد عناصر القطر الرئيسي للمصفوفة.

$$A = \begin{bmatrix} \mathbf{7} & 8 & 9 & 9 \\ 7 & \mathbf{8} & 9 & 9 \\ 4 & 5 & \mathbf{8} & 6 \\ 7 & 8 & 9 & \mathbf{9} \end{bmatrix}$$

```
>> diag (A)
```