را يه ها درر MATLAB به سادگي ايجاد مي شوند.ساده ترين و ابتدايي ترين روش تايپ تمام مقادير بين دو [] مي باشد:

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

اما براي توليد آرايه هاي بزرگتر استفاده از روش بالا بسيار وقت گير است.MATLAB چند دستور براي توليد آرايه ها دارد:

linspace(a,b,c)

اين دستور c نقطه با فاصله هاي مساوي در بازه [a,b] را برمي گرداند.

logspace(a,b,c)

اين دستور نيز c نقطه با فواصل لگاريتمي در بازه $10^a,10^b$بر مي گرداند.

a:b:c

مفهوم كلي اين دستور را مي توان به اين صورت بيان كرد؛ از a شروع كن b تا b تا برو جلو تا به c برسي.در اين روش b كه گام حركت نام دارد مي تواند منفي باشد.

با استفاده از دستورات فوق و توابع رياضي مي توان ماتريس هاي متنوعي توليد كرد.

```
>> x=0:.5:2*pi;
>> y=sin(x)
y =
  Columns 1 through 10
         0    0.4794    0.8415    0.9975    0.9093    0.5985    0.1411   -0.3508   -0.7568   -0.9775
  Columns 11 through 13
-0.9589   -0.7055   -0.2794

>> z=tanh(x)
z =
  Columns 1 through 10
         0    0.4621    0.7616    0.9051    0.9640    0.9866    0.9951    0.9982    0.9993    0.9998
  Columns 11 through 13
 0.9999    1.0000    1.0000
```

آرايه ها در MATLAB تنها به آرايه هاي سطري محدود نمي شوند بلكه مي توان آرايه ها ي ستوني نيز معرفي كرد. براي اينكه به MATLAB بفهمانيم كه قصد توليد سطر جديد را داريم از سميكالن(؛) استفاده مي كنيم

```
>> b=[1;3;5;7]
b =
    1
    3
    5
    7
>> b=linspace(1,7,4)'
b =
    1
    3
    5
    7
```

همان طور كه در مثال بالا مشاهده كرديد بااستفاده از عملگر ترانهاده (') يك آرايه سطري را به آرايه ستوني تبديل كرديم.
براي دسترسي به درايه ها از انديس آن استفاده مي شود.به عنوان مثال b(3) سومين درايه آرايه b (يعني 5) را برمي گرداند.
اگر بخواهيم درايه هاي خاصي از يك آرايه را انتخاب كنيم مي توان با استفاده از دو نقطه به آنها دسترسي داشته باشيم.اگر طول آرايه مشخص نباشد مي توان براي دسترسي به آخرين عنصر از end استفاده كرد

```
>> z(end:-3:4)
ans =
   1.0000    0.9998    0.9951    0.9051
```

همچنين مي توان با استفاده از يك آرايه به مقادير يك آرايه ديگر با ترتيب مورد نظر خودمان استفاده كنيم.

```
>> c=[1,5,3,5,1];
>> z(c)
ans =
        0    0.9640    0.7616    0.9640         0
```

همان طور كه مشاهده كرديد با استفاده از آرايه c عناصر اول،پنجم،سوم،پنجم و اول را فراخواني كرديم.

نكته ديگر كه بايد به آن توجه كرد اين است كه انديس يك عدد صحيح مثبت است. اگر كاربر يك عدد غير صحيح و يا منفي را به عنوان انديس وارد كند MATLAB يك پيغام خطا بر مي گرداند.

```
>> b(3.4)
??? Subscript indices must either be real positive integers or logicals.

>> b(-4)
??? Subscript indices must either be real positive integers or logicals.
```

دو آرايه مفروض a و b را مي توان با استفاده از دستورات [a,b] (الحاق سطري)و [a;b] (الحاق ستوني) به يكديگر الحاق كرد و به اين ترتيب آرايه هايي با سطرها و ستون هاي متعدد داشت؛البته در الحاق سطري تعداد سطرها و در الحاق ستوني تعداد ستون هاي دو آرايه بايد برابر باشد.همچنين مي توان تعداد آرايه هاي الحاقي را به طور دلخواه افزايش داد.

```
>> a=[1 2 3];
>> b=[4 5 6];
>> c=[7 8 9];
>> d=[a b],f=[a;b;c]
d =
    1    2    3    4    5    6
f =
    1    2    3
    4    5    6
    7    8    9
```

به اين ترتيب متوجه شديد كا آرايه ها در MATLAB مي توانند داراي سطرها و ستون هاي متعدد باشند.
حال ممكن است اين سوال پيش بيايد كه چگونه مي توان اين آرايه ها را توليد كرد.همان طور كه قبلا ملاحظه نموديد براي معرفي ستون ها از فاصله يا كاما (،) و براي سطر ها از سميكالن (؛) استفاده مي شود.راه ديگر ايجاد سطر اين است كه بعد از معرفي سطر اول با زدن كليد Enter به خط بعد رفته و به معرفي سطر بعد بپردازيد.

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

```
A =
    1     2     3
    6     5     4

>> B= [10 11 12
13 14 15]
B =
   10    11    12
   13    14    15
```

در اين مورد به چند نكته بايد توجه كرد:
تعداد ستون ها در هر سطر بايد برابر باشد در غير اين صورت پيغام خطايي نمايش داده مي شود.
تعداد فواصلي كه براي جدا كردن اعداد به كار مي رود براي MATLAB مهم نيست.يعني MATLAB فضاهاي خالي اضافي را تشخيص داده و حذف مي كند.

محاسبات عددي آرايه ها:

MATLAB عملگرهاي فراواني براي اعمال بر روي آرايه ها دارد.تعدادي از آنها را در مثال هاي زير مشاهده مي كنيد:

```
>> A
A =
    1     2     3
    6     5     4
>> A-2
ans =
   -1     0     1
    4     3     2
>> A*2 + B
ans =
   12    15    18
   25    24    23
>> A + B
ans =
   11    13    15
   19    19    19
```

همان طور كه در بالا ديديد اعمال ساده رياضي را مي توان روي آرايه ها انجام دهيد.A-2 باعث مي شود از تمام درايه هاي آرايه A دو واحد كم شود.همچنين دستور A+B درايه هاي نظير به نظير دو آرايه را جمع مي كند.

ضرب و توان ماتريس ها نيز به سادگي قابل تعريف اند:

```
>> A
A =
    1     2     3
    6     5     4
>> C=[1 2;3 4];
>> C*A
ans =
   13    12    11
   27    26    25
>> C^2
```

```
ans =
    7    10
   15    22
```

حال اگر بخواهيم تك تك درابه ها را به توان n برسانيم يا درايه هاي دو آرايه يا ماتريس را نظير به نظير در هم ضرب يا تقسيم كنيم بايد قبل از اين عملگرها يك نقطه قرار دهيم.به مثال هاي زير توجه كنيد:

```
>> A.*B
ans =
   10    22    36
   78    70    60

>> B.^A
ans =
        10       121      1728
   4826809    537824     50625

>> C.^2
ans =
    1     4
    9    16
```

آرايه هاي استاندارد:

```
>> ones(2,3)
ans =
    1     1     1
    1     1     1
```

ones(n,m) ماتريسي nxm با درايه هاي يك ايجاد مي كند.

```
>> zeros(2,4)
ans =
    0     0     0     0
    0     0     0     0
```

zeros(n,m) ماتريسي nxm با درايه هاي صفر ايجاد مي كند.

```
>> eye(3,4)
ans =
    1     0     0     0
    0     1     0     0
    0     0     1     0
```

اين دستور هم ماتريسيnxm ايجاد مي كند كه درايه هاي روي قطر اصلي آن يك است.اگر اين دستور به صورت eye(n) به كار رود يك ماتريس همانی به وجود مي آيد.

```
>> rand(2,3)
ans =
   0.9501    0.6068    0.8913
   0.2311    0.4860    0.7621
```

اين دستور هم يك ماتريس با درايه هاي اتفاقي بين صفر و يك ايجاد مي كند.

```
>> randperm(8)
ans =
    2     4     1     5     8     6     3     7
```

```
>> randperm(10)
ans =
   10    7    1    8    2    5    9    6    4    3
```
اين دستور اعداد 1 تا n را به صورت اتفاقي در يك بردار سطري قرار مي دهد.

```
>> magic(3)
ans =
    8    1    6
    3    5    7
    4    9    2
```
اين دستور ماتريسي كه به ماتريس جادويي معروف است توليد مي كند.ويژگي اين ماتريس اين است مجموع درايه هاي هر سطر، ستون و قطر با هم برابر است.

توجه به اين نكته لازم است كه دو دستور آخر تنها يك ورودي دارند،اما دستورات اول مي توانند داراي دو آرگومان ورودي باشند.در صورتي كه دستورات اول با يك آرگومان به كار بروند يك ماتريس مربعي n x n ايجاد مي شود.

اندازه ماتريس ها:

در MATLAB دستوراتي وجود دارد كه به وسيله آنها مي توان اطلاعاتي در مورد يك ماتريس از قبيل تعداد سطر،ستون و تعداد كل عناصر آن را به دست آورد.

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

>> size(a)
ans =
     2     3
```
اين دستور همان طور كه از اسم آن مشخص است اندازه (تعداد سطر و ستون) ماتريس را برمي گرداند.در دستوراتي مانند دستور فوق كه 2 خروجي يا بيشتر دارند مي توان هر خروجي را در يك متغير ذخيره كرد براي اين كار به صورت زير عمل مي كنيم:

```
>> [s t]= size (a)
s =
     2
t =
     3

>> length(a)
ans =
     3
```
اين دستور بزرگترين مقدار بين سطر و ستون را برمي گرداند.

```
>> numel(a)
ans =
     6
```
اين دستور هم تعداد عناصر(number of elements) ماتريس را برمي گرداند.

فرض كنيد مي خواهيم عناصر قطر اصلي يك ماتريس را به دست آوريم و يا مجموع آنها را به دست آوريم.2 دستور زير به ما كمك خواهد كرد.

```
>> diag(a)
ans =
      1
      2

>> trace(a)
ans =
      3
```

البته دستور diag كاربردهاي ديگري نيز دارد.به عنوان مثال اگر به صورت diag(a,n) به كار رود قطر n ام ماتريس a را مي دهد.n مي تواند مثبت يا منفي باشد.
در صورتي كه به صورت diag(a) به كار رود و a يك بردار باشد ماتريسي ايجاد مي كند كه قطر اصلي آن بردارa و ساير عناصر آن صفر است.

```
>> c= [1 2 3];

>> diag (c)
ans =
      1    0    0
      0    2    0
      0    0    3

>> minfo(ans)
3 rows  3 cols: regular MATLAB matrix
```

دستور آخر هم اطلاعاتي در مورد ماتريس مورد نظر را برمي گرداند.

ايجاد تغيير در ماتريس ها:

MATLAB با اختصاص يك انديس به هر عضو آرايه راه هاي زيادي را براي ايجاد تغيير در درايه هاي ماتريس ها به وجود مي آورد.

```
>> c = ans
c =
   1    0    0
   0    2    0
   0    0    3

>> c(3,3) =8
c =
   1    0    0
   0    2    0
   0    0    8

>> c(9)=4
c =
   1    0    0
   0    2    0
   0    0    4
```

همان طور كه قبلا گفته شد براي دسترسي به عناصر آرايه ها 2 راه وجود دارد.راه اول با 2 آرگومان ورودي كه سطر و ستون را مشخص مي كنند.راه ديگر استفاده از يك آرگومان كه در اين صورت انديس هر درايه به صورت زير (ماتريس s) تعيين مي شود:

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

>> c(4)=5
c =
    1    5    0
    0    2    0
    0    0    4

>> c(:,1)=7
c =
    7    5    0
    7    2    0
    7    0    4
```

مفهوم اين دستور اين است كه تمام سطرها در ستون اول را مساوي 7 قرار بده.

```
>> c([1 end],[1 end])=8
c =
    8    5    8
    7    2    0
    8    0    8
```

اين دستور نيز مثل دستور بالا عمل مي كند يعني در سطر اول و آخر ستون هاي اول و آخر را مساوي 8 قرار مي دهد.

```
>> c(1,:)= []
c =
    7    2    0
    8    0    8
```

اين دستور هم با مساوي قرار دادن سطر اول با آرايه تهي اين سطر را حذف مي كند.

تغييراتي كه در ماتريس ها مي توان ايجاد كرد تنها به تغيير عناصر آن محدود نمي شود بلكه مي توان ابعاد، ترتيب و جاي درايه ها را تغيير داد.

```
>> a=[1 2 3; 4 5 6;7 8 9];
>> flipud(a)
ans =
      7    8    9
      4    5    6
      1    2    3
```

اين دستور ماتريس را از بالا به پايين مي چرخاند . يعني جاي سطرها را نسبت به سطر وسط عوض مي كند.

```
>> b=fliplr(a)
b =
      3    2    1
      6    5    4
```

```
     9     8     7
```
اين دستور هم مانند دستور فوق است با اين تفاوت كه روي ستون ها عمل مي كند.

```
>> rot90(b)
ans =
     1     4     7
     2     5     8
     3     6     9
```
اين دستور ماتريس را 90 درجه در خلاف حركت عقربه هاي ساعت مي چرخاند.البته اگر اين دستور به صورت rot90(a,n) به كار رود n تعداد چرخش را مشخص مي كند.

```
>> triu(ans)
ans =
     1     4     7
     0     5     8
     0     0     9

>> tril(ans)
ans =
     1     0     0
     0     5     0
     0     0     9
```
اين دو دستور هم ماتريس هاي بالا و پايين مثلثي توليد مي كنند.

```
>> c=horzcat(ans,b)
c =
    1    0    0    3    2    1
    0    5    0    6    5    4
    0    0    9    9    8    7

>> c=vertcat(ans,b)
c =
    1    0    0
    0    5    0
    0    0    9
    3    2    1
    6    5    4
    9    8    7
```
اين دو دستور هم الحاق سطري و ستوني را انجام مي دهد.

```
>> reshape(c,2,9)
ans =
     1     0     6     0     0     5     0     9     4
     0     3     9     5     2     8     0     1     7
```
اگر بخواهيم ابعاد يك ماتريس را تغيير بدهيم مي توانيم از اين دستور استفاده كنيم.البته بايد تعداد المنت ها قبل و بعد از تغيير با هم برابر باشد.

```
>> repmat(ans(:,1),2,4)
ans =
     1     1     1     1
     0     0     0     0
     1     1     1     1
     0     0     0     0
```

اين دستور با تكرار يك ماتريس ماتريسي با ابعاد جديد توليد مي كند.به عنوان مثال اگر دستور به اين صورت باشد MATLAB repmat(a,m,n) ماتريس a را به عنوان يك المنت در نظر گرفته و mxn بار تكرار مي كند.

يكي ديگر از دستورات كاربردي MATLAB دستور(cat(n,a,b است.در اين دستور a و b دو ماتريس و n مشخص كننده جهت الحاق مي باشد.مثال هاي زير بهتر مفهوم مورد نظر را مي رساند.مثال سوم نمونه اي از ماتريس هاي 3 بعدي مي باشد.به نحوه نمايش اين ماتريس ها دقت كنيد.

```
>> d=cat(1,a,b)
d =
    1    2    3
    4    5    6
    7    8    9
    3    2    1
    6    5    4
    9    8    7

>> d=cat(2,a,b)
d =
    1    2    3    3    2    1
    4    5    6    6    5    4
    7    8    9    9    8    7

>> d=cat(3,a,b)

d(:,:,1) =
    1    2    3
    4    5    6
    7    8    9
d(:,:,2) =
    3    2    1
    6    5    4
    9    8    7
```

در زير يكي ديگر از دستورات ايجاد تغيير در ماتريس ها را مشاهده مي كنيد.

```
>> a=[1 2;3 4];
>> b=[2 3];
>> kron(a,b)
ans =
     2    3    4    6
     6    9    8   12

>> kron(b,a)
ans =
     2    4    3    6
     6    8    9   12
```

مثال اخير را مي توان به اين صورت نيز نمايش داد:

```
>> [ b(1)*a , b(2)*a ]

ans =
     2    4    3    6
     6    8    9   12
```

مرتب كردن آرايه ها:

يكي از امكانات مفيد MATLAB وجود دستوري براي مرتب كردن آرايه ها مي باشد.در زير ابتدا يك ماتريس با درايه هاي بين 0 و 20 توليد مي كنيم و سپس با استفاده از دستور sort(a,n) كه n مشخص كننده سطر يا ستون مي باشد آن را مرتب مي كنيم.

```
>> a=fix(20*rand(3,4))
a =
     8     8     7     2
    18    17    16     4
    18     1     0     3

>> sort(a,1)
ans =
     8     1     0     2
    18     8     7     3
    18    17    16     4

>> [sor,pos]=sort(a,2)
sor =
     2     7     8     8
     4    16    17    18
     0     1     3    18
pos =
     4     3     1     2
     4     3     2     1
     3     2     4     1
```

همان طور كه مشاهده مي كنيد در صورتي كه از MATLAB دو خروجي بخواهيم آنگاه دو ماتريس برمي گرداند؛ ماتريس اول همان ماتريس مرتب شده و ماتريس دوم انديس مربوط به درايه هاي مرتب شده را نمايش مي دهد.به عبارت ديگر نشان دهنده مكان درايه قبل از مرتب شدن مي باشد.

در اين گونه دستورات در صورتي كه n توسط كاربر مشخص نشود 2 حالت پيش مي آيد.
1) در صورتي كه ماتريس 2 بعدي يا يك بردار ستوني باشد ستون ها مورد بررسي قرار مي گيرند.
2) در صورتي كه ماتريس يك بردار سطري باشد سطرها بررسي مي شوند.

جستجو در آرايه ها:

در MATLAB دستورات زيادي براي جستجو در آرايه ها، پيدا كردن عناصر خاص و ... وجود دارد. در زير به برخي از آنها اشاره مي شود.

```
>> a=[0 1 0;2 0 3;1 3 4];
>> nnz(a)
ans =
     6

>> find(a)
ans =

    2
    3
    4
    6
```

```
    8
    9

>> nonzeros(a)
ans =
    2
    1
    1
    3
    3
    4
```

این 3 دستور تقریباً با هم در ارتباط اند.دستور اول تعداد درایه هاي غیر صفر را برمي گرداند.دستور دوم اندیس مربوط به این درایه ها و دستور سوم خود درایه هاي غیر صفر ار برمي گرداند.

```
>> all(a)
ans =
     0     0     0

>> all(a,2)
ans =
     0
     0
     1
```

دستور all(a,n) در صورتي که تمام درایه هاي سطر یا ستون (بستگي به n دارد) غیر صفر باشند مقدار یك و در غیر این صورت صفر را برمي گرداند.

در این دستور هم چنانچه n مشخص نشود مانند آنچه گفته شد عمل مي شود.

```
>> any(a,2)
ans =
     1
     1
     1
```

این تابع در صورتي که یکي از المنت ها غیر صفر باشند مقدار یك و در صورتي که همه صفر باشند مقدار صفر را برمي گرداند.

```
>> isempty(a)
ans =
     0
```

این تابع در صورتي که ماتریس یك ماتریس تهي باشد مقدار 1 را بر مي گرداند.

```
>> b=[9 3 0 8];
>> ismember(b,a)
ans =
     0     1     1     0
```

این دستور در صورتي که اعضاي ماتریس b (که مي تواند یك عدد باشد) عضو a نیز باشند مقدار یك و در غیر این صورت مقدار صفر را برمي گرداند.

```
>> unique(a)
ans =
     0
     1
     2
     3
```

4
و اين تابع يك بردار ستوني شامل عناصر غير تكراري a كه به ترتيب صعودي مرتب شده اند را برمي گرداند.

كاربرد در داده هاي آماري:

MATLAB دستورات فراواني براي كار كردن روي داده ها و انجام دادن كارهاي آماري دارد. در زير به مهمترين آنها اشاره مي شود.

```
>> a=fix(20*rand(3,4))
a =
    19     9     9     8
     4    17     0    12
    12    15    16    15
```

يكي از پركاربردترين دستورات 2 دستور زير است كه به ترتيب مينيمم و ماكسيمم هر ستون را به دست مي آورند. در صورتي كه ماتريس يك بردار سطري باشد اين كار را روي سطر انجام مي دهند.

```
>> min(a)
ans =
     4     9     0     8

>> [s,t]=max(a)
s =
    19    17    16    15
t =
     1     2     3     3
```

همان طور كه ملاحظه مي كنيد در صورتي كه مانند دستور sort دو خروجي بخواهيم خروجي دوم مشخص كننده مكان درايه مينيمم يا ماكزيمم خواهد بود.
در صورتي كه اين دستور به صورت max(a,b) (يا min) به كار رود _كه b يك ماتريس با ابعاد ماتريس a يا يك عدد است_ آنگاه خروجي يك ماتريس است با درايه هاي بزرگتر بين a و b.

```
>> b=fix(45*rand(3,4))
b =
    41    18    18    15
    33    42    40    36
     7    41     2     0

>> max(a,b)
ans =
    41    18    18    15
    33    42    40    36
    12    41    16    15
```

همچنين مي توان ميانگين و عضو مياني هر سطر يا ستون را به دست آورد.

```
>> mean(b,2)
ans =
   23.0000
   37.7500
   12.5000

>> median(b,2)
ans =
```

```
   18.0000
   38.0000
    4.5000

>> median(b,1)
ans =
    33    41    18    15
```

به اين نكته دقت كنيد كه تابع median پس از مرتب كردن سطر يا ستون عنصر مياني را برمي گرداند. و در صورتي كه تعداد سطرها يا ستون ها زوج باشد ميانگين 2 عضو وسط را برمي گرداند.

```
>> sum(b)
ans =
    81   101    60    51
```

مجموع هر سطر يا ستون را برمي گرداند.

```
>> cumsum(b)
ans =
    41    18    18    15
    74    60    58    51
    81   101    60    51
```

عملكرد آن مشابه سيگما در رياضي است.يعني مجموع هر درايه با درايه هاي قبل از آن را بر مي گرداند.

```
>> prod(b)
ans =
        9471       30996        1440           0
```

حاصلضرب درايه هاي هر ستون را برمي گرداند.

```
>> cumprod(b)
ans =
          41          18          18          15
        1353         756         720         540
        9471       30996        1440           0
```

اين دستور حاصلضرب هر درايه در درايه هاي ماقبل را بر مي گرداند. از اين دستور مي توان در شبيه سازي فاكتوريل استفاده كرد.

در دستورات مي توان جهت انجام عمليات (سطر يا ستون) را مشخص كرد.در صورتي كه جهت مشخص نشود مانند ديگر دستورها كه در بالا گفته شد عمل مي شود.

يكي ديگر از دستورات كه براي مقايسه دو ماتريس به كار مي رود دستور زير است:

```
>> isequal(a,b)
ans =
     0
```

در صورتي كه دو ماتريس برابر باشند مقدار يك را برمي گرداند.

ماتريس به عنوان ضرايب چند جمله اي:

يك ديگر از كاربردهاي ماتريس ها استفاده از آنها به عنوان ضرايب يك چندجمله اي است.

```
>> x=[2 3 6 10];
>> y=[1 2 3 4];
```

فرض كنيد تعدادي داده آماري داريد و مي خواهيد براي ارتباط دادن آنها با يكديگر تابعي را پيدا كنيد. MATALAB اين كار را به راحتي و به وسيله polyfit(x,y,n) انجام مي دهد.در اينجا x,y داده ها و n مشخص كننده درجه چندجمله اي مورد نظر است.نتيجه اين تابع يك ماتريس است.

```
>> p=polyfit(x,y,3)
p =
    0.0193   -0.3795    2.5298   -2.6964

>> polyval(p,x)
ans =
   1.0000    2.0000    3.0000    4.0000
```

حال فرض كنيد يك چند جمله اي داريم و مي خواهيم مقادير آن را به ازاي مقادير مختلف يك متغير به دست آوريم.براي اين كار از تابع polyval استفاده مي كنيم.

```
>> r=roots(p)
r =
  9.1557 + 4.8026i
  9.1557 - 4.8026i
  1.3040
```

همان طور كه گفته شد مي توان ماتريس ها را به عنوان ضرايب يك چندجمله اي در نظر گرفت.در اين صورت اگرp يك بردار شامل ضرايب چندجمله اي باشد ريشه هاي آن به روش بالا به دست مي آيد.

```
>> p2=poly(r)
p2 =
   1.0000  -19.6154  130.7692 -139.3846
```

اين دستور عكس دستورroots مي باشد.يعني با داشتن ريشه هاي يك چندجمله اي مي توانيد ضرايب آن را به دست آوريد.

در محاسبات رياضي گاهي بايد 2 چندجمله اي را در هم ضرب يا بر هم تقسيم كرد.2 دستور زير به ترتيب اين كارها را انجام مي دهند.

```
>> p1=[2 0 3 -1];
>> p2=[2 4];

>> conv(p1,p2)
ans =
      4     8     6    10    -4

>> deconv(p1,p2)
ans =
      1.0000   -2.0000    5.5000
```

جمع و تفريق چندجمله اي ها نيز به سادگي جمع و تفريق ماتريس ها مي باشد.البته دو ماتريس بايد برابر باشند.

```
>> p3=[3 2 0 1];
>> p1 + p3
ans =
     5     2     3     0
```

ماتريس به عنوان مجموعه:

يك ديگر از كاربردهاي گسترده ماتريس ها در نظر گرفتن آنها به عنوان يك مجموعه مي باشد.

```
>> a=randperm(8)
a =
    4    3    2    6    8    1    5    7

>> b=2:2:8
b =
    2    4    6    8

>> union(a,b)
ans =
    1    2    3    4    5    6    7    8

>> intersect(b,a)
ans =
    2    4    6    8
```

دستور اول اجتماع و دستور دوم اشتراك دو مجموعه را به دست مي آورد.

```
>> setxor(a,b)
ans =
    1    3    5    7
```

اعضايي كه يا فقط در a هستند يا فقط در b.به عبارت ديگر اجتماع منهي اشتراك.(تفاضل متقارن)

```
>> setdiff(a,b)
ans =
    1    3    5    7
```

اعضايي از a كه در b نيستند.

```
>> setdiff(b,a)
ans =
  Empty matrix: 1-by-0
```

اعضايي از b كه در a نيستند.

محاسبات ماتريسي:

در زير به ترتيب معكوس، دترمينان و ترانهاده ماتريس a را محاسبه مي كنيم.

```
>> a=[2 3 -4;0 -4 2;1 -1 5]
a =
    2    3   -4
    0   -4    2
    1   -1    5

>> inv(a)
ans =
    0.3913    0.2391    0.2174
   -0.0435   -0.3043    0.0870
   -0.0870   -0.1087    0.1739

>> det(a)
```

```
ans =
       -46

>> a'
ans =
      2     0     1
      3    -4    -1
     -4     2     5
```

مديريت فايل ها و متغيرها:

MATLAB چند دستور براي آگاهي كاربر از متغيرها و فايل هاي موجود دارد كه در زير به آنها اشاره مي شود.

```
>> what
M-files in the current directory F:\MATLAB6_5\work

bub_dew     cumprod2    exm         mostafa     prod2       size2       sum2
calculate   cumsum2     moadel      name        seri        star
```

اين تابع M-file هاي موجود در دايركتوري جاري را نمايش مي دهد.براي تغيير دايركتوري مي توانيد همانند سيستم عامل dos از دستور cd استفاده كنيد.

```
>> who
Your variables are:
a    ans  b    p    p2   pos  r    sor  x    y

>> whos
  Name      Size                Bytes  Class

  a         3x3                    72  double array
  ans       1x5                    40  double array
  b         1x4                    32  double array
  p         1x4                    32  double array
  p2        1x4                    32  double array
  pos       3x4                    96  double array
  r         3x1                    48  double array (complex)
  sor       3x4                    96  double array
  x         1x4                    32  double array
  y         1x4                    32  double array

Grand total is 61 elements using 512 bytes
```

يكي از دستورات جالب MATLAB كه بيشتر جنبه سرگرمي دارد دستورwhy است.اين دستور را امتحان كنيد.

```
>> why
Some hamster insisted on it.
```