

را په ها درر MATLAB به سادگي ايجاد مي شوندد. ساده ترين و ابتدائي 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 شوندد بلکه مي توان آرايه هاي 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 که در مثال بالا مشاهده کردید با استفاده از عملگر ترانهاده (') يك آرایه سطري را به آرایه ستوني تبدیل کردیم.

برای دسترسی به درایه ها از اندیس آن استفاده می شود. به عنوان مثال b(3) سومین درایه آرایه b (یعنی 5) را برمی گرداند.

اگر بخواهیم درایه های خاصی از يك آرایه را انتخاب کنیم می توان با استفاده از دو نقطه به آنها دسترسی داشته باشیم. اگر طول آرایه مشخص نباشد می توان برای دسترسی به آخرین عنصر از end استفاده کرد

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

همچنین می توان با استفاده از يك آرایه به مقادیر يك آرایه دیگر با ترتیب مورد نظر خودمان استفاده کنیم.

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

همان طور که مشاهده کردید با استفاده از آرایه c عناصر اول، پنجم، سوم، پنجم و اول را فراخوانی کردیم.

نکته دیگر که باید به آن توجه کرد این است که اندیس يك عدد صحیح مثبت است. اگر کاربر يك عدد غیر صحیح و یا منفی را به عنوان اندیس وارد کند MATLAB يك پیغام خطا بر می 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 آرایه مفروض a و b را می توان با استفاده از دستورات [a,b] (الحاق سطري) و [a;b] (الحاق ستوني) به یکدیگر الحاق کرد و به این ترتیب آرایه هایی با سطرها و ستون های متعدد داشت؛ البته در الحاق سطري تعداد سطرها و در الحاق ستوني تعداد ستون های دو آرایه باید برابر باشد. همچنین می توان تعداد آرایه های الحاقی را به طور دلخواه افزایش 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 این ترتیب متوجه شدید که آرایه ها در MATLAB می توانند دارای سطرها و ستون های متعدد باشند. حال ممکن است این سوال پیش بیاید که چگونه می توان این آرایه ها را تولید کرد. همان طور که قبلاً ملاحظه نمودید برای معرفی ستون ها از فاصله یا کاما (,) و برای سطرها از سمیکالن (!) استفاده می شود. راه دیگر ایجاد سطر این است که بعد از معرفی سطر اول با زدن کلید Enter به خط بعد رفته و به معرفی سطر بعد پردازید.

```
>> 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 این مورد به چند نکته باید توجه کرد: تعداد ستون ها در هر سطر باید برابر باشد در غیر این صورت پیغام خطایی نمایش داده می شود. تعداد فواصلی که برای جدا کردن اعداد به کار می رود برای MATLAB مهم نیست. یعنی MATLAB فضاها را خالی اضافی را تشخیص داده و حذف می کند.

محاسبات عددی آرایه ها:

MATLAB عملگرهای فراوانی برای اعمال بر روی آرایه ها دارد. تعدادی از آنها را در مثال های زیر مشاهده می کنید:

```

>> 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 که در بالا دیدید اعمال ساده ریاضی را می توان روی آرایه ها انجام دهید. $A-2$ باعث می شود از تمام درایه های آرایه A دو واحد کم شود. همچنین دستور $A+B$ درایه های نظیر به نظیر دو آرایه را جمع می کند.

ضرب و توان ماتریس ها نیز به سادگی قابل تعریف 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 بخواهیم تک تک درایه ها را به توان n برسانیم یا درایه های دو آرایه یا ماتریس را نظیر به نظیر در هم ضرب یا تقسیم کنیم باید قبل از این عملگرها یک نقطه قرار دهیم. به مثال های زیر توجه کنید:

```
>> 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
```

آرایه های استاندارد:

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

$\text{ones}(n,m)$ ماتریسی $n \times m$ با درایه های یک ایجاد می کند.

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

$\text{zeros}(n,m)$ ماتریسی $n \times m$ با درایه های صفر ایجاد می کند.

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

این دستور هم ماتریسی $n \times m$ ایجاد می کند که درایه های روی قطر اصلی آن یک است. اگر این دستور به صورت $\text{eye}(n)$ به کار رود یک ماتریس همبندی به وجود می آید.

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

این دستور هم یک ماتریس با درایه های اتفاقی بین صفر و یک ایجاد می کند.

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

```
>> randperm(10)
```

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

این دستور اعداد 1 تا n را به صورت اتفاقی در یک بردار سطری قرار می دهد.

```
>> magic(3)
```

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

این دستور ماتریسی که به ماتریس جادویی معروف است تولید می کند. ویژگی این ماتریس این است مجموع درایه های هر سطر، ستون و قطر با هم برابر است.

توجه به این نکته لازم است که دو دستور آخر تنها یک ورودی دارند، اما دستورات اول می توانند دارای دو آرگومان ورودی باشند. در صورتی که دستورات اول با یک آرگومان به کار بروند یک ماتریس مربعی $n \times n$ ایجاد می شود.

اندازه ماتریس ها:

در MATLAB دستوراتی وجود دارد که به وسیله آنها می توان اطلاعاتی در مورد یک ماتریس از قبیل تعداد سطر، ستون و تعداد کل 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
```

این دستور همان طور که از اسم آن مشخص است اندازه (تعداد سطر و ستون) ماتریس را برمی گرداند. در دستوراتی مانند دستور فوق که 2 خروجی یا بیشتر دارند می توان هر خروجی را در یک متغیر ذخیره کرد برای این کار به صورت زیر عمل می کنیم:

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

```
s =  
2
```

```
t =  
3
```

```
>> length(a)
```

```
ans =  
3
```

این دستور بزرگترین مقدار بین سطر و ستون را برمی گرداند.

```
>> numel(a)
```

```
ans =  
6
```

این دستور هم تعداد عناصر (number of elements) ماتریس را برمی گرداند.

فرض کنید می خواهیم عناصر قطر اصلی یک ماتریس را به دست آوریم و یا مجموع آنها را به دست آوریم. دستوری زیر به ما کمک خواهد کرد.

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

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

البته دستور `diag` کاربردهای دیگری نیز دارد. به عنوان مثال اگر به صورت `diag(a,n)` به کار رود قطر n ام ماتریس a را می دهد. n می تواند مثبت یا منفی باشد. در صورتی که به صورت `diag(a)` به کار رود و a یک بردار باشد ماتریسی ایجاد می کند که قطر اصلی آن بردار a و سایر 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 اطلاعاتی در مورد ماتریس مورد نظر را برمی گرداند.

ایجاد تغییر در ماتریس ها:

MATLAB با اختصاص یک اندیس به هر عضو آرایه راه های زیادی را برای ایجاد تغییر در درایه های ماتریس ها به وجود می 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 که قبلا گفته شد برای دسترسي به عناصر آرایه ها 2 راه وجود دارد. راه اول با آرگومان ورودی که سطر و ستون را مشخص می کنند. راه دیگر استفاده از يك آرگومان که در این صورت اندیس هر درایه به صورت زیر (ماتریس S) تعیین می 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 این دستور این است که تمام سطرها در ستون اول را مساوی 7 قرار بده.

```
>> c([1 end],[1 end])=8
```

```
c =  
 8  5  8  
 7  2  0  
 8  0  8
```

این دستور نیز مثل دستور بالا عمل می کند یعنی در سطر اول و آخر ستون های اول و آخر را مساوی 8 قرار می دهد.

```
>> c(1,:)= []
```

```
c =  
 7  2  0  
 8  0  8
```

این دستور هم با مساوی قرار دادن سطر اول با آرایه تهی این سطر را حذف می کند.

تغییراتی که در ماتریس ها می توان ایجاد کرد تنها به تغییر عناصر آن محدود نمی شود بلکه می توان ابعاد، ترتیب و جای درایه ها را تغییر داد.

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

```
>> flipud(a)
```

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

این دستور ماتریس را از بالا به پایین می چرخاند . یعنی جای سطرها را نسبت به سطر وسط عوض می کند.

```
>> b=fliplr(a)
```

```
b =  
 3  2  1  
 6  5  4
```

9 8 7

این دستور هم مانند دستور فوق است با این تفاوت که روی ستون‌ها عمل می‌کند.

```
>> rot90(b)
```

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

این دستور ماتریس را 90 درجه در خلاف حرکت عقربه‌های ساعت می‌چرخاند. البته اگر این دستور به صورت $\text{rot90}(a,n)$ به کار رود n تعداد چرخش را مشخص می‌کند.

```
>> triu(ans)
```

```
ans =  
    1    4    7  
    0    5    8  
    0    0    9
```

```
>> tril(ans)
```

```
ans =  
    1    0    0  
    0    5    0  
    0    0    9
```

این دو دستور هم ماتریس‌های بالا و پایین مثلثی تولید می‌کنند.

```
>> 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
```

این دو دستور هم الحاق سطری و ستونی را انجام می‌دهد.

```
>> reshape(c,2,9)
```

```
ans =  
    1    0    6    0    0    5    0    9    4  
    0    3    9    5    2    8    0    1    7
```

اگر بخواهیم ابعاد یک ماتریس را تغییر بدهیم می‌توانیم از این دستور استفاده کنیم. البته باید تعداد المنت‌ها قبل و بعد از تغییر 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
```

این دستور با تکرار يك ماتریس ماتریسی با ابعاد جدید تولید می کند. به عنوان مثال اگر دستور به این صورت باشد `repmat(a,m,n)` MATLAB ماتریس `a` را به عنوان يك المنت در نظر گرفته و `mxn` بار تکرار می کند.

يکي ديگر از دستورات کاربردي MATLAB دستور `cat(n,a,b)` است. در این دستور `a` و `b` دو ماتریس و `n` مشخص کننده جهت الحاق می باشد. مثال هاي زیر بهتر مفهوم مورد نظر را می رساند. مثال سوم نمونه اي از ماتریس هاي 3 بعدي می باشد. به نحوه نمایش این ماتریس ها دقت کنید.

```
>> 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 زیر يکي ديگر از دستورات ایجاد تغییر در ماتریس ها را مشاهده می کنید.

```
>> 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 می توان به این صورت نیز نمایش داد:

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

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

مرتب کردن آرایه ها:

یکی از امکانات مفید MATLAB وجود دستوری برای مرتب کردن آرایه ها می باشد. در زیر ابتدا یک ماتریس با درایه های بین 0 و 20 تولید می کنیم و سپس با استفاده از دستور `sort(a,n)` که `n` مشخص کننده سطر یا ستون می باشد آن را مرتب می کنیم.

```
>> 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 که مشاهده می کنید در صورتی که از MATLAB دو خروجی بخواهیم آنگاه دو ماتریس برمی گرداند؛ ماتریس اول همان ماتریس مرتب شده و ماتریس دوم اندیس مربوط به درایه های مرتب شده را نمایش می دهد. به عبارت دیگر نشان دهنده مکان درایه قبل از مرتب شدن می باشد.

در این گونه دستورات در صورتی که `n` توسط کاربر مشخص نشود 2 حالت پیش می آید.
(1) در صورتی که ماتریس 2 بعدی یا یک بردار ستونی باشد ستون ها مورد بررسی قرار می گیرند.
(2) در صورتی که ماتریس یک بردار سطری باشد سطرها بررسی می شوند.

جستجو در آرایه ها:

در MATLAB دستورات زیادی برای جستجو در آرایه ها، پیدا کردن عناصر خاص و ... وجود دارد. در زیر به برخی از آنها اشاره می 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 های غیر صفر را برمی گرداند. دستور دوم اندیس مربوط به این درایه ها و دستور سوم خود درایه های غیر صفر را برمی گرداند.

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

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

دستور $all(a,n)$ در صورتی که تمام درایه های سطر یا ستون (بستگی به n دارد) غیر صفر باشند مقدار یک و در غیر این صورت صفر را برمی گرداند.

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

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

این تابع در صورتی که یکی از المنت ها غیر صفر باشند مقدار یک و در صورتی که همه صفر باشند مقدار صفر را برمی گرداند.

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

این تابع در صورتی که ماتریس یک ماتریس تهی باشد مقدار 1 را بر می گرداند.

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

این دستور در صورتی که اعضای ماتریس b (که می تواند یک عدد باشد) عضو a نیز باشند مقدار یک و در غیر این صورت مقدار صفر را برمی گرداند.

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

4

و این تابع يك بردار ستونی شامل عناصر غير تكراري a که به ترتیب صعودی مرتب شده اند را برمی گرداند.

کاربرد در داده های آماری:

MATLAB دستورات فراوانی برای کار کردن روی داده ها و انجام دادن کارهای آماری دارد. در زیر به مهمترین آنها اشاره می شود.

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

یکی از پرکاربردترین دستورات 2 دستور زیر است که به ترتیب مینیمم و ماکسیمم هر ستون را به دست می آورند. در صورتی که ماتریس يك بردار سطری باشد این کار را روی سطر انجام می دهند.

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

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

همان طور که ملاحظه می کنید در صورتی که مانند دستور sort دو خروجی بخواهیم خروجی دوم مشخص کننده مکان درایه مینیمم یا ماکزیمم خواهد بود. در صورتی که این دستور به صورت $\max(a,b)$ (یا \min) به کار رود که b يك ماتریس با ابعاد ماتریس a یا يك عدد است_ آنگاه خروجی يك ماتریس است با درایه های بزرگتر بین 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
```

همچنین می توان میانگین و عضو میانی هر سطر یا 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 این نکته دقت کنید که تابع median پس از مرتب کردن سطر یا ستون عنصر میانی را برمی گرداند. و در صورتی که تعداد سطرها یا ستون ها زوج باشد میانگین 2 عضو وسط را برمی گرداند.

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

مجموع هر سطر یا ستون را برمی گرداند.

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

عملکرد آن مشابه سیگما در ریاضی است. یعنی مجموع هر درایه با درایه های قبل از آن را بر می گرداند.

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

حاصلضرب درایه های هر ستون را برمی گرداند.

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

این دستور حاصلضرب هر درایه در درایه های ماقبل را بر می گرداند. از این دستور می توان در شبیه سازی فاکتوریل استفاده کرد.

در دستورات می توان جهت انجام عملیات (سطر یا ستون) را مشخص کرد. در صورتی که جهت مشخص نشود مانند دیگر دستورها که در بالا گفته شد عمل می شود.

یکی دیگر از دستورات که برای مقایسه دو ماتریس به کار می رود دستور زیر است:

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

در صورتی که دو ماتریس برابر باشند مقدار یک را برمی گرداند.

ماتریس به عنوان ضرایب چند جمله ای:

یک دیگر از کاربردهای ماتریس ها استفاده از آنها به عنوان ضرایب یک چندجمله ای است.

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

فرض کنید تعدادی داده آماری دارید و می خواهید برای ارتباط دادن آنها با یکدیگر تابعی را پیدا کنید. MATALAB این کار را به راحتی و به وسیله $\text{polyfit}(x,y,n)$ انجام می دهد. در اینجا x,y داده ها و n مشخص کننده درجه چندجمله ای مورد نظر است. نتیجه این تابع یک ماتریس 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 کنید یک چند جمله ای داریم و می خواهیم مقادیر آن را به ازای مقادیر مختلف یک متغیر به دست آوریم. برای این کار از تابع polyval استفاده می کنیم.

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

همان طور که گفته شد می توان ماتریس ها را به عنوان ضرایب یک چندجمله ای در نظر گرفت. در این صورت اگر p یک بردار شامل ضرایب چندجمله ای باشد ریشه های آن به روش بالا به دست می آید.

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

این دستور عکس دستور roots می باشد. یعنی با داشتن ریشه های یک چندجمله ای می توانید ضرایب آن را به دست آورید.

در محاسبات ریاضی گاهی باید 2 چندجمله ای را در هم ضرب یا بر هم تقسیم کرد. 2 دستور زیر به ترتیب این کارها را انجام می 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 تفریق چندجمله ای ها نیز به سادگی جمع و تفریق ماتریس ها می باشد. البته دو ماتریس باید برابر باشند.

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

ماتریس به عنوان مجموعه:

یک دیگر از کاربردهای گسترده ماتریس ها در نظر گرفتن آنها به عنوان یک مجموعه می 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 اشتراک دو مجموعه را به دست می آورد.

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

اعضایی که یا فقط در a هستند یا فقط در b . به عبارت دیگر اجتماع منهی اشتراک. (تفاضل متقارن)

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

اعضایی از a که در b نیستند.

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

اعضایی از b که در a نیستند.

محاسبات ماتریسی:

در زیر به ترتیب معکوس، دترمینان و ترانزاده ماتریس a را محاسبه می کنیم.

```
>> 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
```

مدیریت فایل ها و متغیرها:

MATLAB چند دستور برای آگاهی کاربر از متغیرها و فایل های موجود دارد که در زیر به آنها اشاره می 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
```

این تابع M-file های موجود در دایرکتوری جاری را نمایش می دهد. برای تغییر دایرکتوری می توانید همانند سیستم عامل dos از دستور cd استفاده کنید.

```
>> 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

یکی از دستورات جالب MATLAB که بیشتر جنبه سرگرمی دارد دستور why است. این دستور را امتحان کنید.

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