

فصل دوم: آرایه و ماتریس ها

معرفی چند تابع برای کار با ماتریس ها:

The screenshot displays the MATLAB environment. The Command Window shows the execution of two functions: `zeros(3,3)` and `ones(3,4)`. The Workspace window shows variables `a`, `ans`, `b`, `x`, `y`, and `z` with their respective dimensions and values. The Command History window shows a list of recent commands and their execution times.

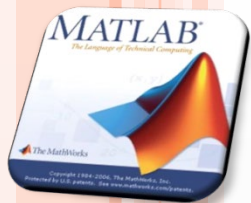
```
>> zeros(3,3)
ans =
     0     0     0
     0     0     0
     0     0     0

>> ones(3,4)
ans =
     1     1     1     1
     1     1     1     1
     1     1     1     1

>>
```

Name	Value
a	<4x3 double>
ans	<3x4 double>
b	<3x4 double>
x	<4x4 double>
y	[1 3 2 4; 5 7 6 8]
z	[1 2 3 4; 5 6 7 8]

Time	Command
6/12/03 8:16 PM	%--
6/12/03 9:02 PM	%--
6/12/03 10:02 PM	%--
6/12/03 10:41 PM	%--
6/13/03 10:51 PM	%--
6/16/03 4:50 AM	%--
6/16/03 8:59 PM	%--
6/16/03 9:04 PM	%--
6/19/03 6:07 AM	%--



فصل دوم: آرایه و ماتریس ها

معرفی چند تابع برای کار با ماتریس ها:

The screenshot shows the MATLAB environment with the following components:

- Command Window:** Contains the following commands and their outputs:

```
>> eye(3,3)

ans =

     1     0     0
     0     1     0
     0     0     1

>> fix(10*rand(1,10))

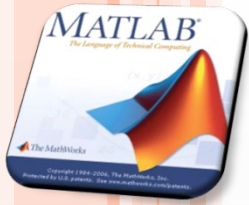
ans =

     9     2     6     4     8     7     4     0     8     4

>>
```
- Workspace:** A table listing variables and their values:

Name	Value
a	<4x3 double>
ans	[9 2 6 4 8 7 4 0 8]
b	<3x4 double>
x	<4x4 double>
y	[1 3 2 4; 5 7 6 8]
z	[1 2 3 4; 5 6 7 8]
- Command History:** A list of recent commands and their execution times, such as '6/12/03 8:16 PM'.

www.Feach.Foghraree.ir



فصل سوم: چند جمله ای ها

کار با چند جمله ای ها:

یک چند جمله ای در MATLAB به صورت یک بردار سطری که مولفه های آن ضرایب چند جمله ای به ترتیب نزولی هستند معرفی می شود. برای مثال چند جمله ای $p(x) = x^2 - 2x + 5$ در MATLAB به شکل زیر معرفی می گردد:

```
» p=[1 0 -2 5];
```

ریشه های یک چند جمله ای

ریشه های یک چند جمله ای را می توانید به صورت زیر بدست آورد:

```
» r=roots(p)
```

```
r =
```

```
-2.0946
```

```
1.0473 + 1.1359i
```

```
1.0473 - 1.1359i
```

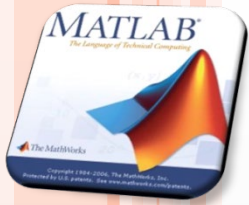
48

با دانستن ریشه های معادله می توانید ضرایب چند جمله ای مربوطه را محاسبه نمایید:

```
» p2=poly(r)
```

```
p2 =
```

```
1.0000 0.0000 -2.0000 5.0000
```



فصل سوم: چند جمله ای ها

۳-۳ ضرب و تقسیم چند جمله ایها

برای ضرب و تقسیم چند جمله ایها می توانید توابع `conv` و `deconv` را بکار ببرید. چند جمله ایهای $a(x)=x^2+x+1$ و $b(x)=x-1$ را در نظر بگیرید. حاصلضرب این دو چند جمله ای به طریق زیر بدست می آید:

```
» a=[1 1 1]; b=[1 -1];
```

```
» c=conv(a,b)
```

```
c =
```

```
1 0 0 -1
```

و تقسیم a/b نیز به صورت زیر قابل محاسبه است:

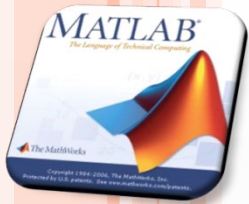
```
» [q,r]=deconv(a,b)
```

```
q =
```

```
1 2
```

```
r =
```

```
0 0 3
```



فصل سوم: چند جمله ای ها

کار با چند جمله ای ها: مشتق چند جمله ای ها:

۳-۴ مشتق چند جمله ای

مشتق چند جمله ای را می توانید با بکار بردن تابع `polyder` محاسبه کنید.

```
» c=polyder(a)
```

```
c =
```

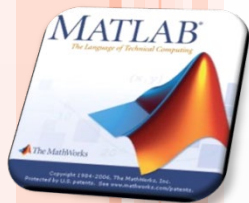
```
2 1
```

مشتق حاصلضرب دو چند جمله ای $(a \times b)$ را می توانید به صورت زیر بدست آورید:

```
» d=polyder(a,b)
```

```
d =
```

```
3 0 0
```



فصل سوم: چند جمله ای ها

۳-۴ برازش منحنی چند جمله ای

تابع polyfit ضرایب بهترین چند جمله ای را پیدا می کند که از میان مجموعه نقاط داده شده عبور می نماید. به عنوان مثال مجموعه نقاط زیر را در نظر بگیرید:

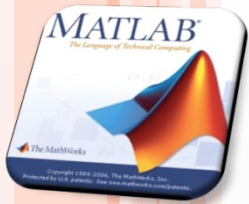
```
» x=[1 2 3 4 5];  
» y=[5.5 43.1 128 290.7 498.4];
```

دستور زیر ضرایب بهترین چند جمله ای درجه سوم را محاسبه می کند که از بین نقاط فوق می گذرد:

```
» p=polyfit(x,y,3)  
p =  
-0.1917 31.5821 -60.3262 35.3400
```

حال می توانید برای مقایسه منحنی محاسبه شده و داده های اولیه را در یک نمودار رسم کنید:

```
» x2=1:1:5;  
» y2=polyval(p,x2);  
» plot(x,y,'o',x2,y2)
```



فصل چهارم: ساختار حلقه ها

تصمیم گیری و روند کنترل:

این توابع برای تصمیم گیری برای انجام کاری و یا انجام به دفعات به کار می رود .

این توابع به چند دسته تقسیم می شوند :

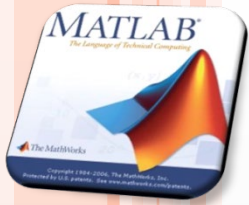
توابع شرطی

- if** این تابع در صورت صادق بودن شرط وارد شده دستورات معین شده را یکبار اجرا میکند .
- while** این تابع تا وقتی که شرط وارد شده برقرار است دستورات وارد شده انجام خواهد شد و تعداد تکرار مهم نیست .
- switch** این دستور تصمیم گیری در میان چندین موضوع همسان را بر عهده دارد تا حدی شبیه به دستور if می باشد .

توابع چرخه ای

- for** بوسیله این دستور می توانیم قسمتی از برنامه را به تعداد معلوم تکرار نماییم .
- while** توضیح این دستور در بالا داده شد که تکرار به دفعات نامتناهی را می توان انجام داد .

در پایان همه این دستورات دستور end استفاده می شود که نشان دهنده پایان چرخه می باشد .



فصل چهارم: ساختار حلقه ها

تصميم گيري و روند كنترل:

if.....end

گفتيم اين تابع اين تابع براي اجاي قسمتي از برنامه در صورت صادق بودن شرط وارد شده به كار مي رود .

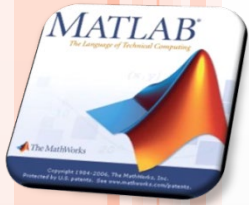
شكل اين دستور بدینصورت مي باشد :

```
if شرط تعيين شده  
    دستورات وارد شده  
    :  
    :  
    :  
end
```

اگر شرط صادق کند دستورات داخل این عبارت اجرا می شود .

ولی اگر شرط صادق نکند کل عبارت if نادیده گرفته می شود و اجرای دستورات به بعد از خط end

منتقل می شود .



فصل چهارم: ساختار حلقه ها

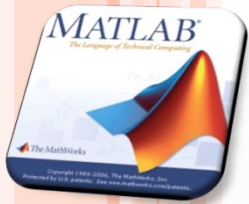
تصمیم گیری و روند کنترل:

else در غیر این صورت

این دستور در داخل عبارت if استفاده می گردد و در صورتی که شرط وارد شده صدق نکند دستورات وارد شده اجرا می شوند .

به مثال زیر توجه فرمائید(در m-file نوشته می شود):

```
a=input('please enter a number : ');  
if a == 2  
    disp('your entered number is 2');  
else  
    disp(' your entered number is not 2');  
end
```



فصل چهارم: ساختار حلقه ها

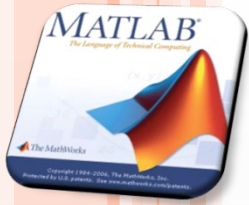
تصمیم گیری و روند کنترل:

elseif چند شرط همزمان

ممکن است در یک زمان چند شرط را همزمان و پشت سر هم وارد کنیم این کار را بوسیله elseif انجام می دهیم .

به مثال زیر توجه فرمائید(در m-file نوشته می شود):

```
a=input('please enter a number :')
if a == 2
    disp('your entered number is 2')
elseif a == 3
    disp('your entered number is 3')
elseif a == 4
    disp('your entered number is 4')
else
    disp('your entered number is not 2 3 4 ')
end
```



فصل چهارم: ساختار حلقه ها

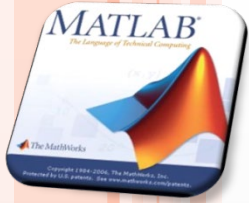
تصمیم گیری و روند کنترل:

Switchcaseend

گفتیم که این دستور برای تصمیم گیری برای حالت های مختلف بکار می رود .

برای مثال آخرین مثال عبارت قبل را با استفاده از switch مینویسیم :

```
a=input('please enter a number :');
switch a
    case 2
        disp('your entered number is 2')
    case 3
        disp('your entered number is 3')
    case 4
        disp('your entered number is 4')
    otherwise
        disp('your entered number is not 2 3 4 ')
end
```



فصل چهارم: ساختار حلقه ها

تصميم گيرى و روند كنترل:

```
for.....end
```

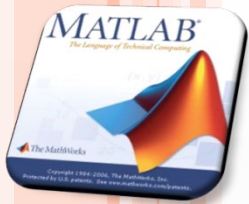
تكرار به تعداد معين

برای تعیین تعداد این تابع باید یک ماتریس سطری با همان تعداد درایه معرفی کرد که این ماتریس را عموماً با (:) مشخص می کنند که ماتریسی با فاصله درایه های یک می سازد .

```
for i=1:10
```

دستورات

```
end
```



فصل چهارم: ساختار حلقه ها

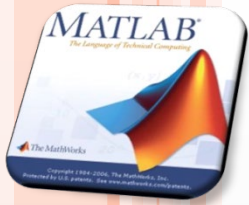
تصميم گيري و روند كنترل:

تكرار تا صادق بودن شرط

while.....end

در اين دستور دقيقا مانند if شرطی وارد می شود ولی نحوه كنترل روند برنامه بدین صورت است كه تا وقتی كه شرط صادق باشد دستورات داخل عبارت while تكرر خواهند شد . (اگر برنامه شتابهی داشته باشد كه همیشه شرط برقرار باشد سيستم در شرط باقی خواهد ماند .)

```
a=input('please enter a number :');
while a ~=1
    if a>1
        a=a-1;
    end
    if a<1
        a=a+1;
    end
end
disp(a);
end
```



فصل چهارم: ساختار حلقه ها

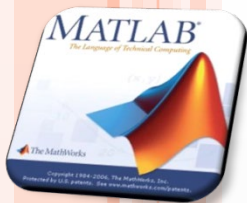
تصميم گيرى و روند كنترل:

break

با اجراى اين دستور مسير برنامه در هر حالتى كه باشد كاملا از داخل حلقه خارج خواهد شد .

به اين مثال توجه فرماييد :

```
for i=1:10
    if i==5
        disp(' in this level path of running was jump out of statement');
        break;
    end
end
i
end
```



فصل چهارم: ساختار حلقه ها

تصمیم گیری و روند کنترل:

منطق در شرط

گاهی اوقات ممکن است لازم باشد چندین شرط را ترکیب کنیم . این عمل با استفاده از عبارت های منطقی قابل انجام است .

اگر بخواهیم دو شرط در یک زمان برقرار باشد از (&) و در صورتی که بخواهیم یکی از دو یا چند شرط صدق کند از (|) استفاده می کنیم .

به مثال های زیر توجه فرمایید :

```
a=input('please enter a number');
b=input('please enter a number');
if a>0 & b>0
    disp('a and b are positive')
elseif a<0 & b<0
    disp('a and b are negative')
elseif a>0 | b<0
    disp('a is positive or b is negative ')
elseif a<0 | b>0
    disp('a is negative or b is positive')
end
```