

MATLAB

(*MATRIX*
LABORATORY)



1

disp KOMUTU İLE EKRANA METİN YAZDIRMA

```
>> disp ('Merhaba')
```

```
Merhaba
```

disp KOMUTU İLE EKRANA SAYISAL DEĞER YAZDIRMA

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

```
>> disp(a)
```

```
1    2
```

```
3    4
```

DISP KOMUTU İLE EKRANA METİNLERİ VE SAYISAL DEĞERLERİ BİRLİKTE YAZDIRMA

Command Window

```
>> tahmin=input('Lutfen Bir Sayi Giriniz: ');  
Lutfen Bir Sayi Giriniz: 12  
>> disp(['Girdiginiz Sayi: ' num2str(tahmin) ' dir.']);  
Girdiginiz Sayi: 12 dir.
```

- MATLAB'in *num2str* fonksiyonu kendisine parametre olarak aldığı bir sayıyı bir karakter dizisine (string) çevirir. (22 sayısı '22' olur.)

FPRINTF KOMUTU İLE EKRANA BİLGİ YAZDIRMA

fprintf('Ekran Basılacak Açıklama %X \n', *deger*);

Burada %X Kısımında Kullanabileceğimiz Seçenekler :

- %c** : *deger* in **tek bir karakter** olduğunu gösterir.
- %s** : *deger* in **bir karakter dizisi (string)** olduğunu gösterir.
- %d** : *deger* in **bir tamsayı** olduğunu gösterir.
- %f** : *deger* in **bir ondalıklı sayı** olduğunu gösterir.
(Noktadan sonra 6 basamak)
- %e** : *deger* i **üstel** formatta yazdırır.
- %g** : *deger* i mümkün olan en kompakt forma sokar.
(%f veya %e seçeneklerinden hangisi kısaysa onu yazdırır)

Diğer yandan :

- \n** : İmleci bir alt satırın başına götürür. (**n**, **newline**)
- \t** : İmleci bir TAB kadar sağa kaydırır.

FPRINTF KULLANIMINA ÖRNEKLER

Command Window

```
>> karakter='d';
>> isim='deniz';
>> tamsayi=25;
>> ondalikliSayi=3.1416;
>> fprintf('Tanimlanan Karakter = %c',karakter);
Tanimlanan Karakter = d>>
>> fprintf('Tanimlanan Karakter Dizisi = %s \n', isim);
Tanimlanan Karakter Dizisi = deniz
>> fprintf('Tanimlanan Tamsayi = %d \n',tamsayi);
Tanimlanan Tamsayi = 25
>> fprintf('Tanimlanan Ondalikli Sayi = %f \n',ondalikliSayi);
Tanimlanan Ondalikli Sayi = 3.141600
>> fprintf('Tanimlanan Ondalikli Sayi = %g \n',ondalikliSayi);
Tanimlanan Ondalikli Sayi = 3.1416
>> fprintf('Tamsayi = %d ve Ondalikli Sayi = %f \n',tamsayi,ondalikliSayi);
Tamsayi = 25 ve Ondalikli Sayi = 3.141600
```

AŞAĞIDAKİ 2 KOMUT AYNI İŞİ YAPAR

```
disp(' Merhaba ');
```

```
fprintf(' Merhaba \n');
```

- **disp** komutu ekrana çıktı verdikten sonra bir alt satıra otomatik olarak atlar. **fprintf** komutunu bir alt satıra götürebilmek için ise **\n** kullanılmalıdır.
- Ayrıca **disp** komutu satır veya sütun vektörleri ile matrisleri ekrana kolayca yazdırabilirken aynı işlemi **fprintf** ile yapabilmek daha çok işlem gerektirmektedir.

MATLAB'DE PROGRAMLAMA

MATLAB'de programlama genel olarak iki yolla yapılır:

- Komut satırında (inline) programlama
- m-dosyaları ile (m-files) programlama
 - Düzyazı (script) m-dosyaları ile programlama
 - Fonksiyon (function) m-dosyaları ile programlama

m-dosyaları oluşturabilmek için ise bir metin düzenleyicisine (editor) ihtiyaç vardır.

KOMUT SATIRINDAKI DEĞİŞKENLERİN KAYDEDİLMESİ

MATLAB komut satırında **save** komutu kullanılırsa o esnada bellekte bulunan değişkenleri, istenilen dosya ismiyle ve uzantısı **.mat** olacak şekilde kaydeder.(Örnek: **sayilar.mat**)

```
>> a=1
```

```
a =
```

```
    1
```

```
>> b=2
```

```
b =
```

```
    2
```

```
>> save sayilar
```

Yukarıda a ve b sayıları sayilar.mat dosyası olarak kaydedilmiştir.

load komutu : Diskte saklı bir dosya içindeki değişkenleri tekrar belleğe yükler.

```
>> load sayilar
```

```
>> who
```

```
Your variables are:
```

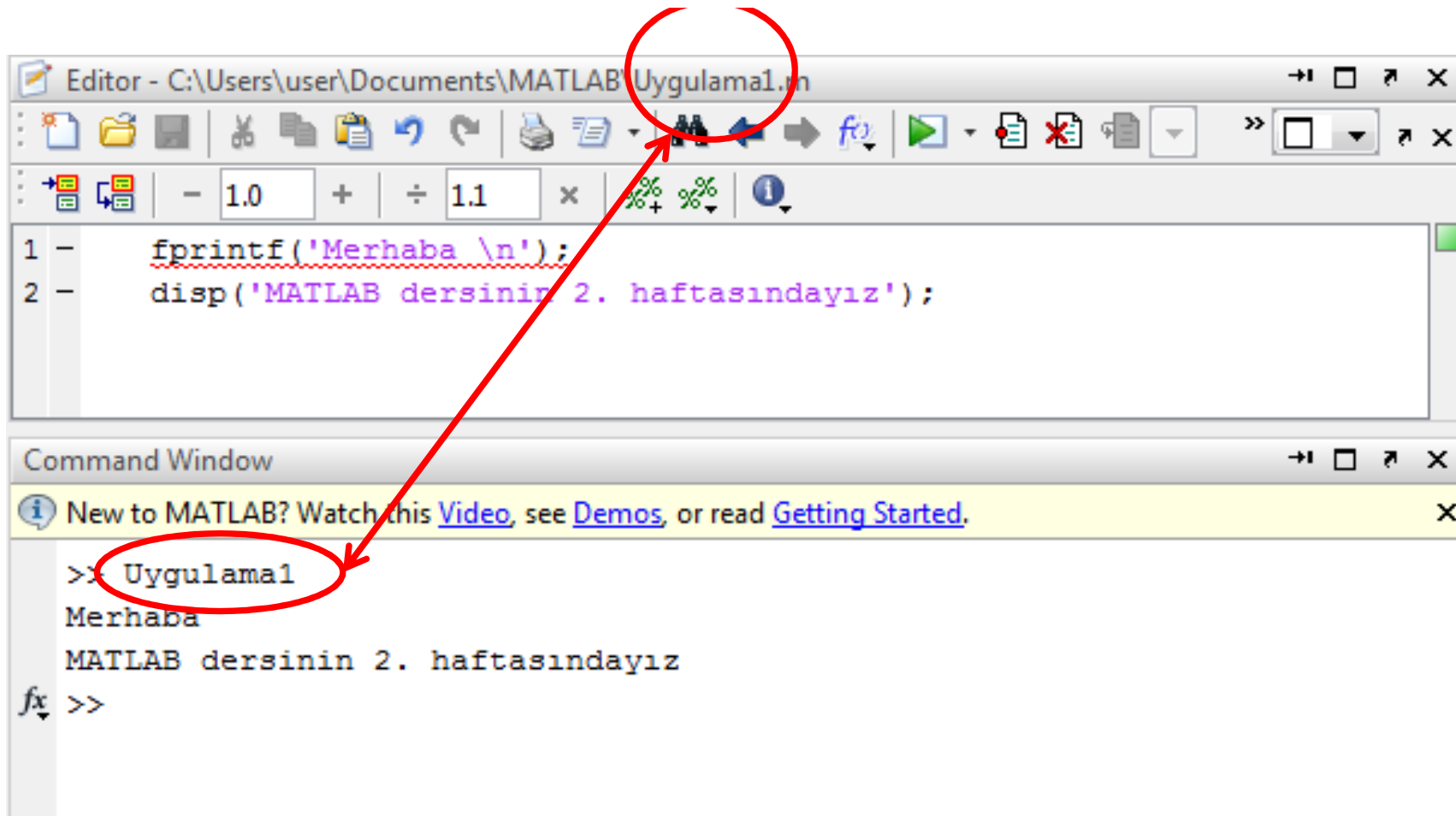
```
a  b
```


M-DOSYALARININ GEREKLİLİĞİ

➤ Bir hesaplamayı gerçekleştirmek için yazılacak birçok komut dizisi, komut penceresinden tek tek girmek yerine bir dosyada saklanır ve daha sonra bu dosya çalıştırılarak bu komutlar icra edilir.

M-dosyalarının Kullanılmasında Dikkat Edilecek Hususlar:

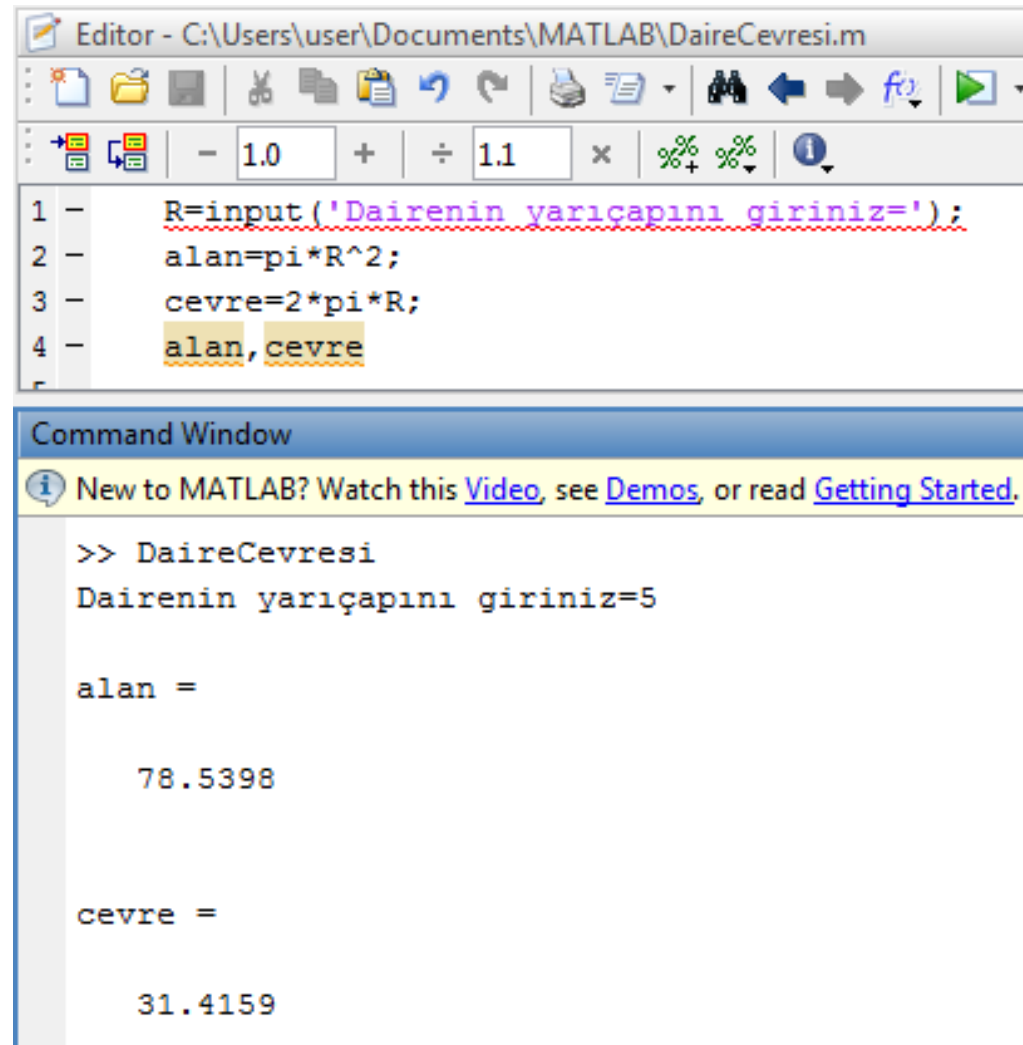
- **M-dosyası** Current Folder altında bulunmalıdır.
- **M-dosyasına** isim verilirken kesinlikle Türkçe karakter kullanılmamalıdır
- **M-dosyasına** isim verilirken MATLAB'in hazır komutları (**pi, exp, sin vs.**) verilmemelidir.



- Programı yazma işlemi bittikten sonra ve programı çalıştırmadan önce muhakkak kaydediniz.
- Komut satırında, değişken ve dosya adlarının ilk birkaç karakterini yazdıktan sonra TAB tuşuna basarak MATLAB'in bu adları otomatik olarak tamamlamasını sağlayabilirsiniz.

Örnek:

Ekrandan dairenin yarıçapını isteyerek alanını ve çevresini hesaplayan programı M-dosyası kullanarak hazırlayınız



The image shows a MATLAB Editor window titled "Editor - C:\Users\user\Documents\MATLAB\DaireCevresi.m". The script contains the following code:

```
1 - R=input('Dairenin yarıçapını giriniz=');
2 - alan=pi*R^2;
3 - cevre=2*pi*R;
4 - alan,cevre
```

Below the editor is the Command Window, which displays the execution of the script:

```
>> DaireCevresi
Dairenin yarıçapını giriniz=5

alan =

    78.5398

cevre =

    31.4159
```

- Fonksiyon girdileri skaler veya matris olabilir.

```
Command Window
>> x=9
x =
     9
>> sqrt(x)
ans =
     3
>> x=[4, 9, 16]
x =
     4     9    16
>> sqrt(x)
ans =
     2     3     4
```

- Remainder fonksiyonu sonuç olarak bölme işleminden kalanı verir.
- Örneğin $10/3$ işleminde kalan 1'dir.

```
Command Window
>> rem(10,3)
ans =
     1
```

- Bazı fonksiyonlar çoklu sonuç verir. Örneğin **size** fonksiyonu satır ve sütun sayısını verir.

```
Command Window
>> d=[1,2,3; 4,5,6];
>> f=size(d)
f =
     2     3
```

- Çıktıya isim atayabilirsiniz:

```
Command Window
>> d=[1,2,3; 4,5,6];
>> [rows,cols]=size(d)
rows =
      2
cols =
      3
```

Değişken isimleri keyfi olabilir.

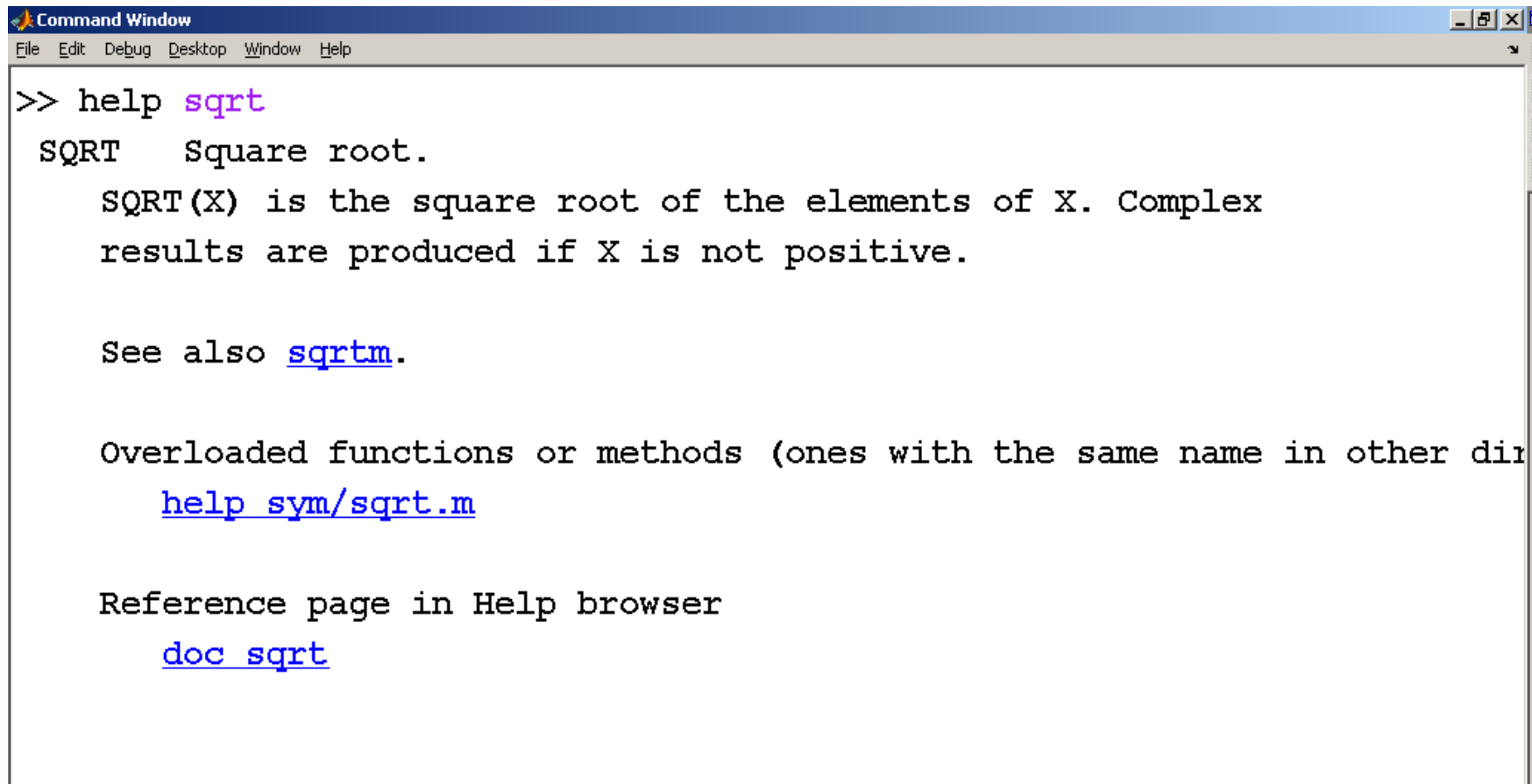
İÇİÇE FONKSİYONLAR

```
Command Window
>> x=2
x =
    2
>> g=sqrt(sin(x))
g =
    0.9536
>> |
```

YARDIM

- Yapmak istediğiniz birçok şey için fonksiyonlar vardır.
- help özelliğini kullanarak bu fonksiyonların ne olduğunu ve nasıl kullanıldığını öğrenebilirsiniz.
 - Komut penceresinden
 - Menü çubuğundaki Help seçimi ile

KOMUT PENCERESİNDEN



A screenshot of the MATLAB Command Window. The title bar reads "Command Window". The menu bar includes "File", "Edit", "Debug", "Desktop", "Window", and "Help". The command prompt shows the user has entered ">> help sqrt". The output text is as follows:

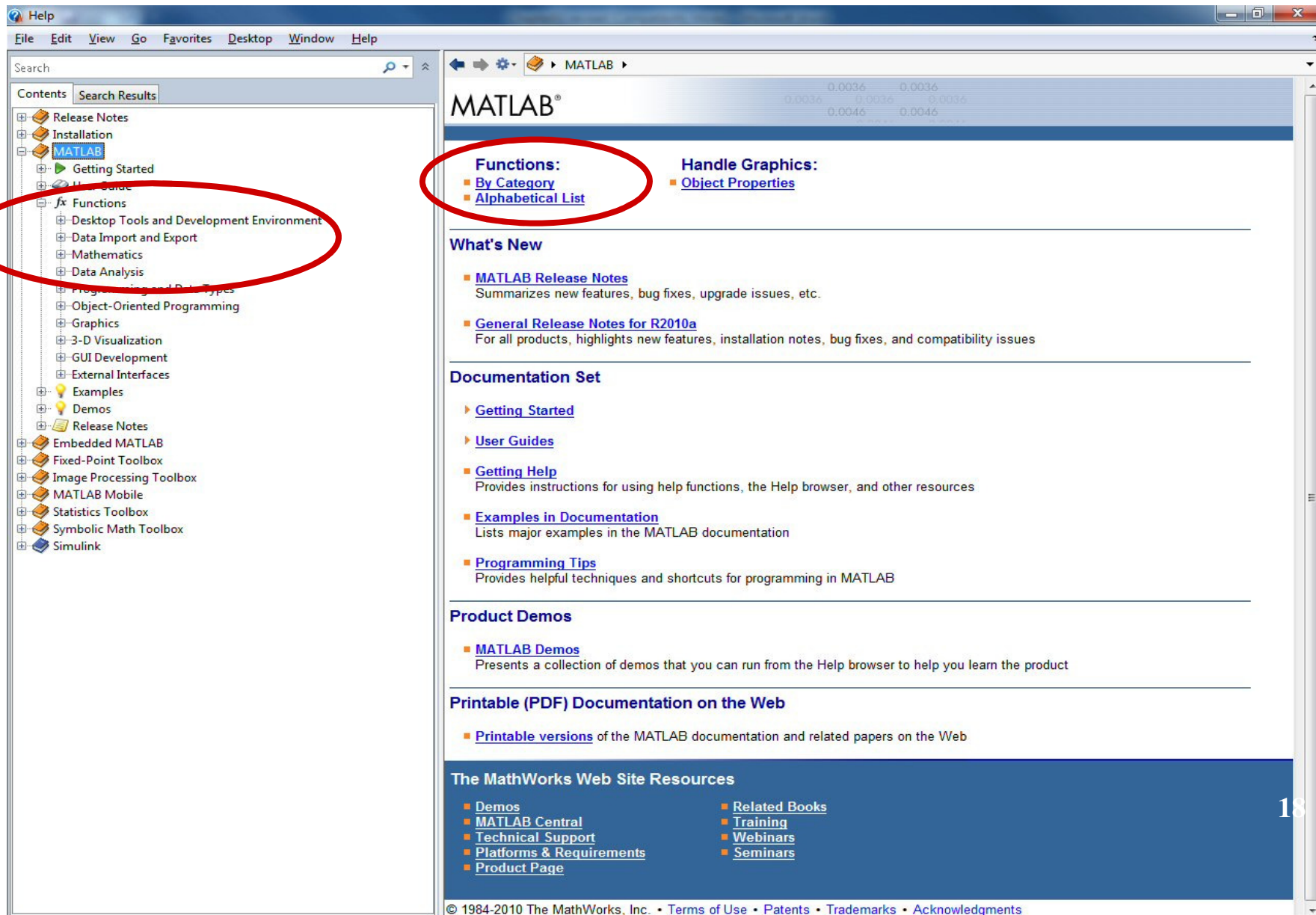
```
>> help sqrt
SQRT    Square root.
        SQRT(X) is the square root of the elements of X. Complex
        results are produced if X is not positive.

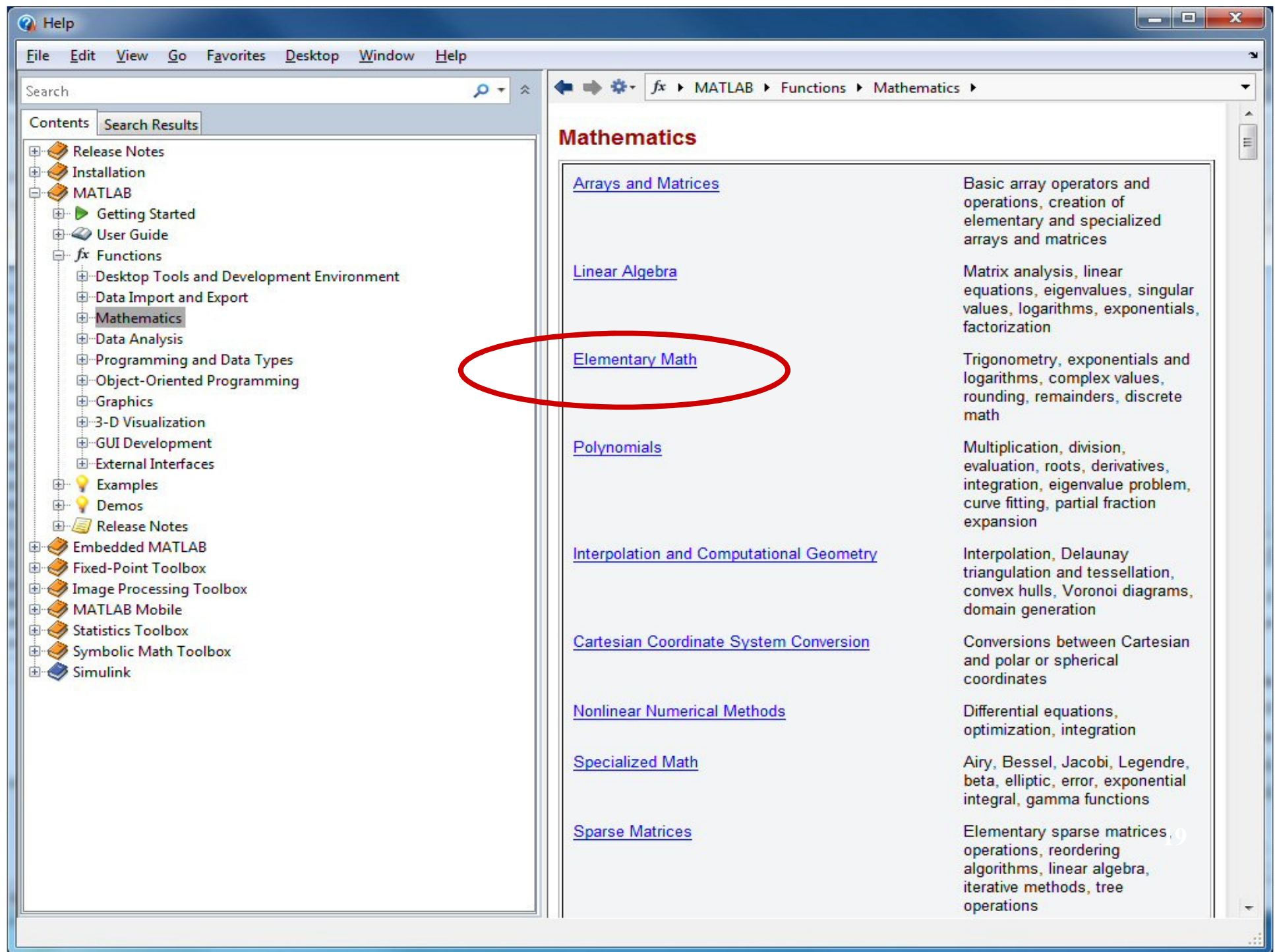
        See also sqrtm.

        Overloaded functions or methods (ones with the same name in other dir
        help sym/sqrt.m

        Reference page in Help browser
        doc sqrt
```

HELP MENÜSÜNDEN





Help

File Edit View Go Favorites Desktop Window Help

fx MATLAB Functions Mathematics Elementary Math Exponential sqrt

There are other functions or methods named [sqrt](#): [simulink/sqrt](#), [fixedpoint/sqrt](#)

sqrt
Square root

Syntax
 $B = \text{sqrt}(X)$

Description
 $B = \text{sqrt}(X)$ returns the square root of each element of the array X . For the elements of X that are negative or complex, $\text{sqrt}(X)$ produces complex results.

Tips
See [sqrtm](#) for the matrix square root.

Examples

```
sqrt((-2:2)')  
ans =  
    0 + 1.4142i  
    0 + 1.0000i  
    0  
    1.0000  
    1.4142
```

See Also
[nthroot](#) | [realsqrt](#) | [sqrtm](#)

Contents Search Results

- User's Guide
- fx Functions
 - Desktop Tools and Development Environment
 - Data Import and Export
 - Mathematics
 - Arrays and Matrices
 - Linear Algebra
 - Elementary Math
 - Trigonometric
 - Exponential
 - fx exp - Exponential
 - fx expm1 - Compute $\exp(x)-1$ accurately
 - fx log - Natural logarithm
 - fx log10 - Common (base 10) logarithm
 - fx log1p - Compute $\log(1+x)$ accurately
 - fx log2 - Base 2 logarithm and dissection
 - fx nextpow2 - Next higher power of 2
 - fx nthroot - Real n th root of real number
 - fx pow2 - Base 2 power and scale factor
 - fx power - Array power
 - fx reallog - Natural logarithm for real numbers
 - fx realpow - Array power for real numbers
 - fx realsqrt - Square root for nonnegative real numbers
 - fx **sqrt** - Square root
 - Complex
 - Rounding and Remainder
 - Discrete Math

○ >> doc sqrt
komutuyla da bu
pencereyi açmak
mümkün

lookfor komutu da adını bilmediğimiz komutları bulmamızda yardımcı olur.

>> lookfor inverse

inverter	- Inverses of Matrices
invhilb	- Inverse Hilbert matrix.
ipermute	- Inverse permute array dimensions.
acos	- Inverse cosine, result in radians.
acosd	- Inverse cosine, result in degrees.
acosh	- Inverse hyperbolic cosine.
acot	- Inverse cotangent, result in radian.
acotd	- Inverse cotangent, result in degrees.
acoth	- Inverse hyperbolic cotangent.
acsc	- Inverse cosecant, result in radian.

•

•

•

•

•

•

BASİT MATEMATİKSEL FONKSİYONLAR

- $\text{abs}(x)$ mutlak değer
- $\text{sign}(x)$ Sayı pozitifse 1 negatise -1 verir
- $\text{exp}(x)$ e^x
- $\log(x)$ doğal logaritma ($\ln(x)$)
- $\log_{10}(x)$ 10 tabanında logaritma

Yuvarlama fonksiyonlari

`round(x)` En yakın tamsayıya

`fix(x)` 0'a doğru yuvarlar

`floor(x)` $-\infty$ 'a doğru yuvarlar

`ceil(x)` ∞ 'a doğru yuvarlar

```
Command Window

>> fix(4.8)
ans =
     4

>> floor(4.8)
ans =
     4

>> ceil(4.8)
ans =
     5

>> fix(-4.8)
ans =
    -4

>> floor(-4.8)
ans =
    -5

>> ceil(-4.8)
ans =
    -4
```

AYRIK MATEMATİK

- `factor(x)` x 'in çarpanlarını verir
- `gcd(x,y)` ortak bölenlerin en büyüğü (greatest common divisor)
- `lcm(x,y)` ortak çarpanların en küçüğü (least common multiple)
- `rats(x)` Sonucu kesirli verir
- `factorial(x)` x faktoriyel
- `nchoosek(n,k)` $N! / K!(N-K)!$
- `primes(x)` x 'e kadar olan asal sayılar
- `isprime(x)` x asal sayı ise 1, değilse 0 verir.

TRİGONOMETRİK FONKSİYONLAR

- $\sin(x)$ sinus (x radyan cinsinden)
- $\text{sin}(d(x))$ sinus(x derece cinsinden)
- $\cos(x)$, $\text{cos}(d(x))$ cosinus
- $\tan(x)$, $\text{tan}(d(x))$ tanjant
- $\cot(x)$, $\text{cot}(d(x))$ kotanjant
- $\text{asin}(x)$, $\text{asind}(x)$ sinus'ün tersi
- $\sinh(x)$ hiperbolik sinüs
- $\text{asinh}(x)$ hyperbolic sinüs'ün tersi

ALIŞTIRMA

Aşağıdakileri hesaplayın (Matematiksel gösterimin, MATLAB gösterimiyle her zaman aynı olmayabileceğini unutmayın)

1. $\sin(2\theta)$ for $\theta = 3\pi$.
2. $\cos(\theta)$ for $0 \leq \theta \leq 2\pi$; let θ change in steps of 0.2π .
3. $\sin^{-1}(1)$.
4. $\cos^{-1}(x)$ for $-1 \leq x \leq 1$; let x change in steps of 0.2.
5. Find the cosine of 45° .
 - a. Convert the angle from degrees to radians, and then use the **cos** function.
 - b. Use the **cosd** function.
6. Find the angle whose sine is 0.5. Is your answer in degrees or radians?
7. Find the cosecant of 60. You may have to use the help function to find the appropriate syntax.

CEVAPLAR

```
1. theta = 3*pi;
   sin(2*theta)
   ans =
   -7.3479e-016

2. theta = 0:0.2*pi:2*pi;
   cos(theta)
   ans =
   Columns 1 through 7
   1.0000 0.8090 0.3090 -0.3090 -0.8090 -1.0000 -0.8090
   Columns 8 through 11
   -0.3090 0.3090 0.8090 1.0000

3. asin(1)
   ans =
   1.5708    This answer is in radians.

4. x = -1:0.2:1;
   acos(x)
   ans =
   Columns 1 through 7
   3.1416 2.4981 2.2143 1.9823 1.7722 1.5708 1.3694
   Columns 8 through 11
   1.1593 0.9273 0.6435 0
```

CEVAPLAR

5. `cos(45*pi/180)`

```
ans =  
    0.7071
```

```
cosd(45)
```

```
ans =  
    0.7071
```

6. `asin(0.5)`

```
ans =  
    0.5236
```

This answer is in radians. You could also find the result in degrees.

```
asind(0.5)
```

```
ans =  
    30.0000
```

7. `csc(60*pi/180)`

```
ans =  
    1.1547
```

or ...

```
cscd(60)
```

```
ans =  
    1.1547
```

VERİ ANALİZİ FONKSİYONLARI

- `max(x)` x vektörünün en büyük değeri
- `min(x)` x vektörünün en küçük değeri
- `mean(x)` ortalama
- `median(x)` medyan değer
- `sum(x)` x vektöründeki elemanlar toplamı
- `prod(x)` elemanların çarpımı
- `sort(x)` x vektöründeki elemanların sıralanması
- `sortrows(x)` ilk sütunu sıralar
- `std(x)` standart sapma
- `var(x)` varyans

MATLAB 7.12.0 (R2011a)

File Edit Debug Desktop Window Help

C:\Users\Holly\Documents\MATLAB

Shortcuts How to Add What's New

Current Folder

« MATLAB »

Name

- html
- a.asv
- a.m
- ackermann.m
- adding_machine.asv
- adding_machine.fig
- adding_machine.m
- alternating_harmonic_serie...
- alternating_harmonic_serie...
- alternating_harmonic_serie...
- arcsin.m
- blast_off_gui.asv
- blast_off_gui.fig
- blast_off_gui.m
- carbon_diffusing.m
- CEU_fac_salary_schedule.asv
- CEU_fac_salary_schedule.m
- CEU_salary_schedule_2008...
- copper_vacancies.jpg
- createfigure.m
- createfigure1.m
- cruise_vacation_compariso...
- cruise_vacation_compariso...

Details

Select a file to view details

Command Window

```
>> x=[1,5,3];
>> max(x)
ans =
     5
>> x=[1,5,3;2,4,6]
x =
     1     5     3
     2     4     6
>> max(x)
ans =
     2     5     6
fx >>
```

x matris ise max her sütun için en büyük değeri verir.

Workspace

Name	Value
ans	[2,5,6]
g	0.9536
x	[1,5,3;2,4,6]

Command History

```
ceil(4.8)
fix(-4.8)
floor(-4.8)
ceil(-4.8)
clc
x=[1,5,3];
max(x)
x=[1,5,3;2,4,6]
max(x)
```

Start

OVR

Command Window

```
>> x=[1,5,3];
```

```
>> [a,b]=max(x)
```

```
a =
```

```
5
```

← max değer

```
b =
```

```
2
```

← İndis no

fx >> |

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

```
x =
```

```
1
```

```
5
```

```
3
```

```
2
```

```
4
```

```
6
```

```
>> [a,b]=max(x)
```

```
a =
```

```
2
```

```
5
```

```
6
```

← max değerlerin vektörü

```
b =
```

```
2
```

```
1
```

```
2
```

← Satır numaralarının vektörü

Command Window

```
>> x=[1 6 3 9 4]
x =
     1     6     3     9     4
>> sort(x)
ans =
     1     3     4     6     9
>> sort(x, 'descend')
ans =
     9     6     4     3     1
```

sort fonksiyonu default olarak artan sırada sıralar. Azalan sırada sıralamak için 2. girdi alanına 'descend' yazmak gerekir.

Command Window

```
>> x=[1 3; 10 2; 3 1; 82 4; 5 5]
```

```
x =
```

1	3
10	2
3	1
82	4
5	5

```
>> sort(x)
```

```
ans =
```

1	1
3	2
5	3
10	4
82	5

MATLAB sütun baskındır, 2 boyutlu matriste sort kullandığımız zaman her sütun artan sırada sıralanır.

```
>> sortrows(x)
```

```
ans =
```

1	3
3	1
5	5
10	2
82	4

```
>> sortrows(x,2)
```

```
ans =
```

3	1
10	2
1	3
82	4
5	5

```
>> sortrows(x,-2)
```

```
ans =
```

5	5
82	4
1	3
10	2
3	1

Azalan sırada sıralamak için sütun numarası önüne – işareti koymak yeterlidir.

MATRİS BOYUTUNU BELİRLEME

- `size(x)` satır ve sütun sayısı
- `length(x)` en büyük boyut
- `numel(x)` elemanların sayısı

```
Command Window

>> x=[1,5,3;2,4,6]
x =
     1     5     3
     2     4     6

>> size(x)
ans =
     2     3

>> length(x)
ans =
     3

>> numel(x)
ans =
     6
```

VARYANS VE STANDART SAPMA

- std(x)

- var(x)

$$\sigma^2 = \frac{\sum_{k=1}^N (x_k - \mu)^2}{N - 1}$$

Standart sapma varyansın kare köküdür. Veri değerlerinin yayılımının özetlenmesi için kullanılan bir ölçüdür. Yani veri değerlerinin aritmetik ortalamadan farklarının karelerinin toplamının veri_sayısı-1'e bölümünün kare köküdür.

RASTGELE SAYILAR

- `rand(n)`
 - $n \times n$ boyutunda, 0-1 aralığında sayılardan oluşan matris oluşur
- `rand(n,m)`
 - $n \times m$ boyutunda 0-1 aralığında sayılardan oluşan matris döndürür

```
Command Window
>> rand(3)
ans =
    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
>> rand(1,3)
ans =
    0.9649    0.1576    0.9706
```

- $M(:)$

- 2 boyutlu bir matrisi tek bir sütuna dönüştürür.

```
>> x=rand(3)
```

```
x =
```

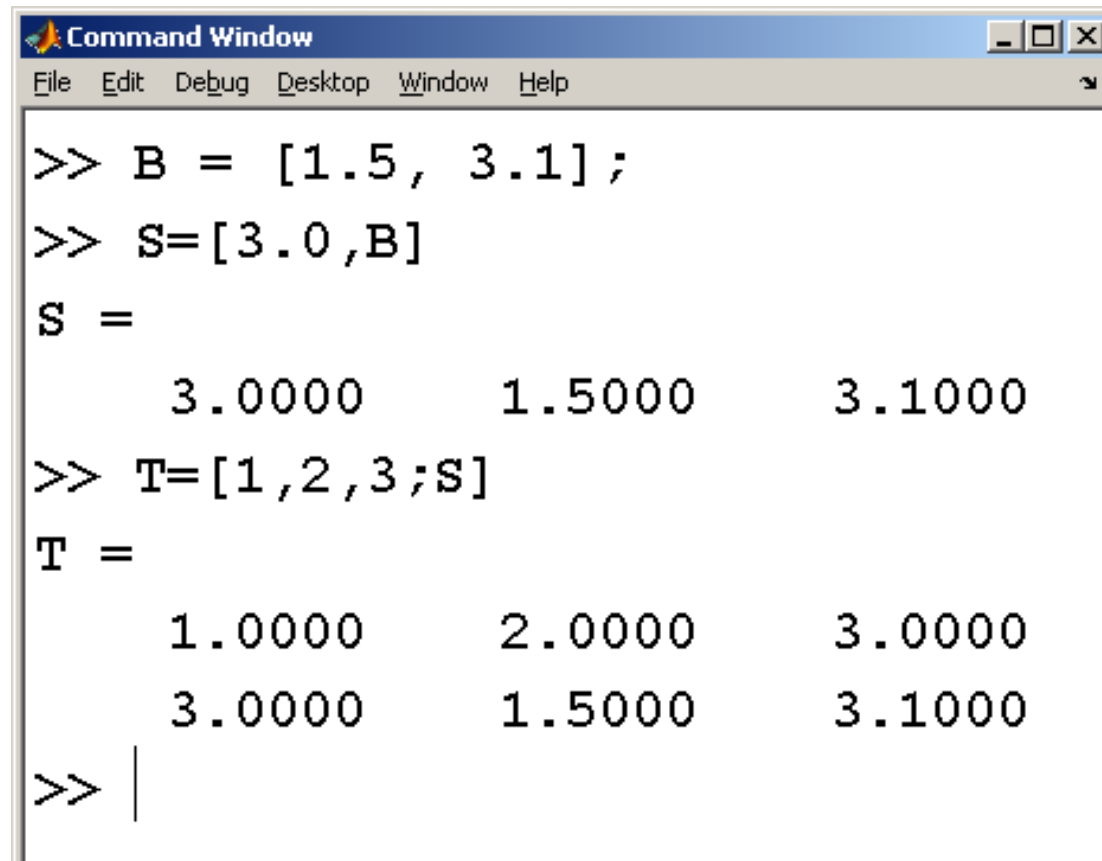
```
0.9655    0.7421    0.6876  
0.1264    0.2924    0.1623  
0.9729    0.5911    0.5620
```

```
>> x(:)
```

```
ans =
```

```
0.9655  
0.1264  
0.9729  
0.7421  
0.2924  
0.5911  
0.6876  
0.1623  
0.5620
```

Diğer matrisi bileşen olarak kullanıp yeni bir matris tanımlamak mümkün.

A screenshot of the MATLAB Command Window. The window has a title bar with the MATLAB logo and the text 'Command Window'. Below the title bar is a menu bar with 'File', 'Edit', 'Debug', 'Desktop', 'Window', and 'Help'. The main area of the window contains the following text:

```
>> B = [1.5, 3.1];  
>> S=[3.0,B]  
S =  
    3.0000    1.5000    3.1000  
>> T=[1,2,3;S]  
T =  
    1.0000    2.0000    3.0000  
    3.0000    1.5000    3.1000  
>> |
```

The screenshot shows the MATLAB Command Window interface. The title bar reads 'Command Window'. The menu bar includes 'File', 'Edit', 'Debug', 'Desktop', 'Window', and 'Help'. The command history shows the creation of matrix B, its use in matrix S, and then matrix T which concatenates a row vector with S. The final state shows the prompt '>>' followed by a vertical bar, indicating the user is ready for the next command.

ELEMAN EKLEMEK

```
>> S
S =
    3.0000    1.5000    3.1000
>> S(2)=1.0
S =
    3.0000    1.0000    3.1000
>> S(4)=5.5
S =
    3.0000    1.0000    3.1000    5.5000
>> S(8)=9.5
S =
Columns 1 through 4
    3.0000    1.0000    3.1000    5.5000
Columns 5 through 8
         0         0         0    9.5000
```

Dizinin boyutları dışında bir indise eleman eklendiğinde arada kalan değerler 0 olur.


```
>> D=1:4
```

```
D =
```

```
1 2 3 4
```

```
>> D(5:10)=10:5:35
```

```
D =
```

```
1 2 3 4 10 15 20 25 30 35
```

```
>> A=[3 8 1 24];
```

```
>> B=4:3:16;
```

```
>> C=[A B]
```

```
C =
```

```
3 8 1 24 4 7 10 13 16
```

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

```
E =
```

```
1 2 3 4
```

```
5 6 7 8
```

```
>> E(3,:)=10:4:22]
```

```
E =
```

```
1 2 3 4
```

```
5 6 7 8
```

```
10 14 18 22
```

```
Command Window
File Edit Debug Desktop Window Help

>> H=1:8
H =
    Columns 1 through 7
         1         2         3         4         5         6         7
    Column 8
         8
>> |
```

default aralık 1

```
Command Window
File Edit Debug Desktop Window Help

>> time=0.0:0.5:2.0
time =
    Columns 1 through 4
         0    0.5000    1.0000    1.5000
    Column 5
         2.0000
>>
```

Aralık 0.5
olarak
belirtilmiş

```
Command Window
File Edit Debug Desktop Window Help

>> M=[1 2 3 4 5
      2 3 4 5 6
      3 4 5 6 7];

>> x=M(:,1)
x =
    1
    2
    3

>> y=M(:,4)
y =
    4
    5
    6

>> z=M(1,:)
z =
    1    2    3    4    5
```

1. sütun

4. sütun

1. satır

```
Command Window
File Edit Debug Desktop Window Help

>> M
M =
     1     2     3     4     5
     2     3     4     5     6
     3     4     5     6     7

>> w=M(2:3,:)
w =
     2     3     4     5     6
     3     4     5     6     7
```

Tüm sütunların 2. ve 3. satırları

```
Command Window
File Edit Debug Desktop Window Help

>> M
M =
     1     2     3     4     5
     2     3     4     5     6
     3     4     5     6     7

>> w=M(2:3, 4:5)
w =
     5     6
     6     7
```

4. ve 5. sütunlarda 2. ve 3. satırlar

```
Command Window
File Edit Debug Desktop Window Help

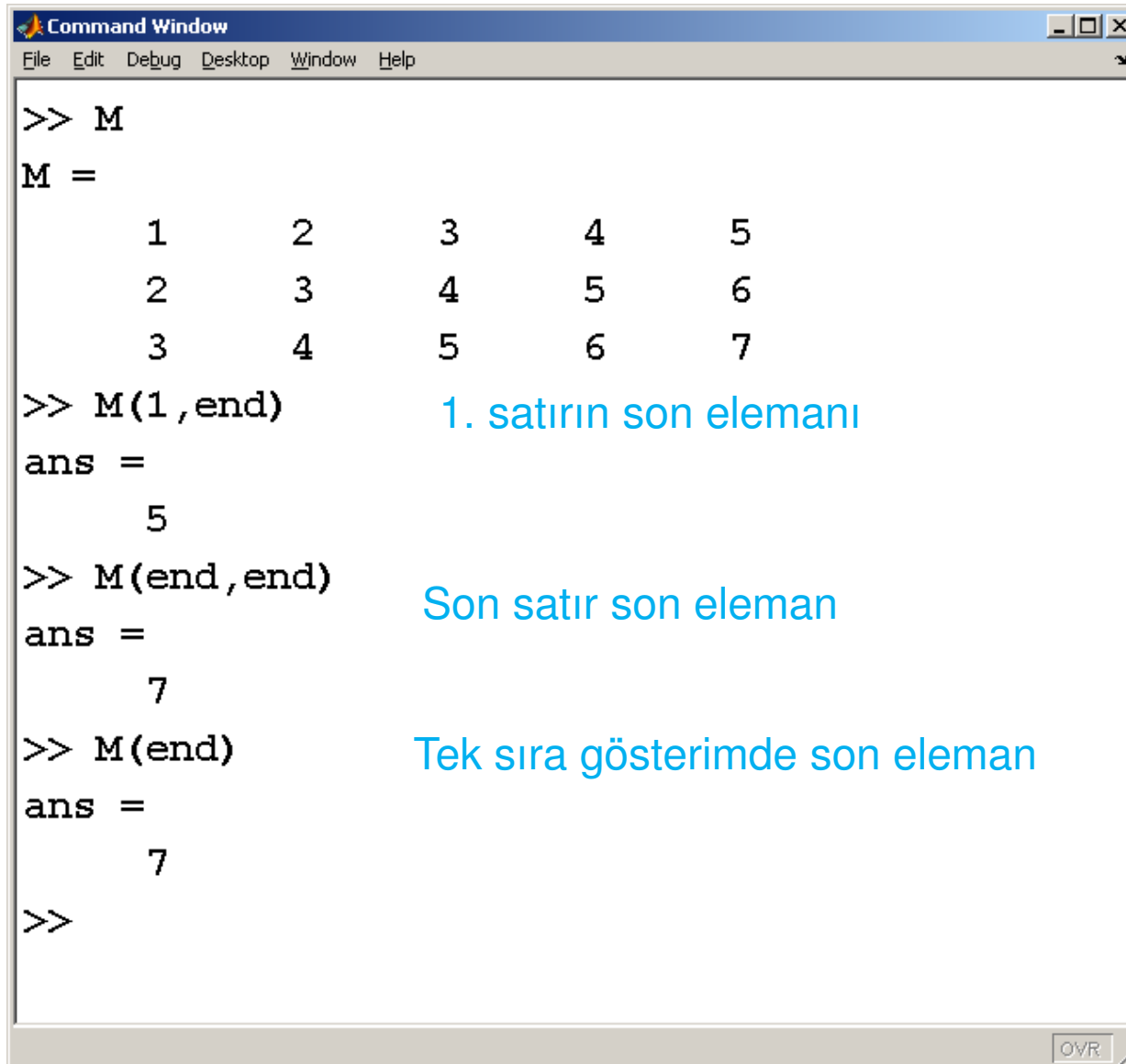
M =
     1     2     3     4     5
     2     3     4     5     6
     3     4     5     6     7

>> M(:)
ans =
     1
     2
     3
     2
     3
     4
     3
     4
     5
     4
     5
     6
     5
     6
     7
```

M(8) ve M(2,3) aynı
elemanı gösterir.

1	4	7	10	13
2	5	8	11	14
3	6	9	12	15

end kelimesi satır veya sütundaki son elemanı gösterir.



```
Command Window
File Edit Debug Desktop Window Help

>> M
M =
     1     2     3     4     5
     2     3     4     5     6
     3     4     5     6     7

>> M(1,end)      1. satırın son elemanı
ans =
     5

>> M(end,end)    Son satır son eleman
ans =
     7

>> M(end)        Tek sıra gösterimde son eleman
ans =
     7

>>
```


İKİ DEĞİŞKENLİ PROBLEMLER

Skaler ve vektörün çarpımını bir vektör verir.

```
Command Window
File Edit Debug Desktop Window Help

>> x=1:5
x =
     1     2     3     4     5
>> y=5;
>> A=x*y
A =
     5    10    15    20    25
>> |
```

İki vektörün çarpılabilmesi için aynı sayıda elemanı olması gerekir.

```
Command Window
File Edit Debug Desktop Window Help

>> x=1:5
x =
     1     2     3     4     5
>> y=1:3
y =
     1     2     3
>> A=x*y
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> A=x.*y
??? Error using ==> times
Matrix dimensions must agree.
```

```
Command Window
File Edit Debug Desktop Window Help

>> x=1:5
x =
     1     2     3     4     5

>> y=linspace(1,3,5)
y =
Columns 1 through 3
     1.0000     1.5000     2.0000
Columns 4 through 5
     2.5000     3.0000

>> A=x.*y
A =
     1     3     6    10    15

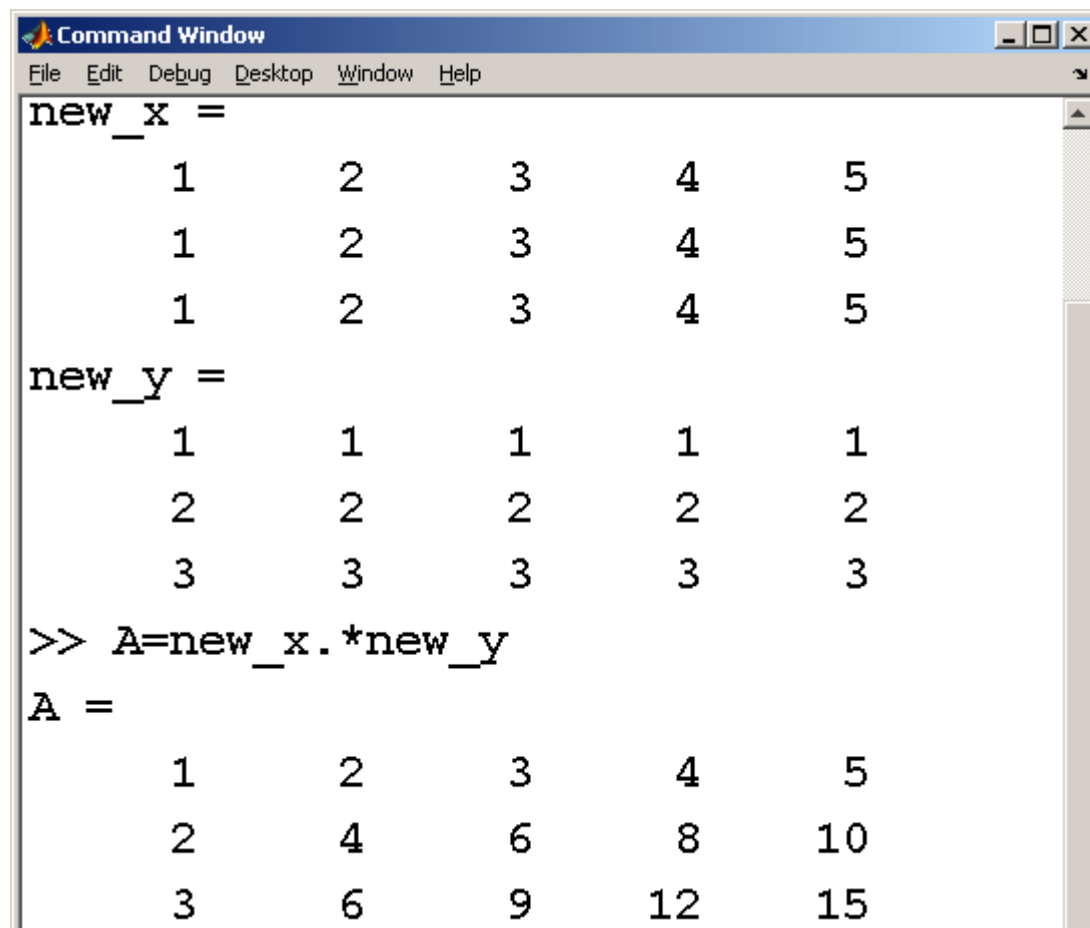
>> |
```

x ve y aynı
boyutta olmalı

meshgrid fonksiyonu iki vektörü, iki boyutlu olacak şekilde eşler.

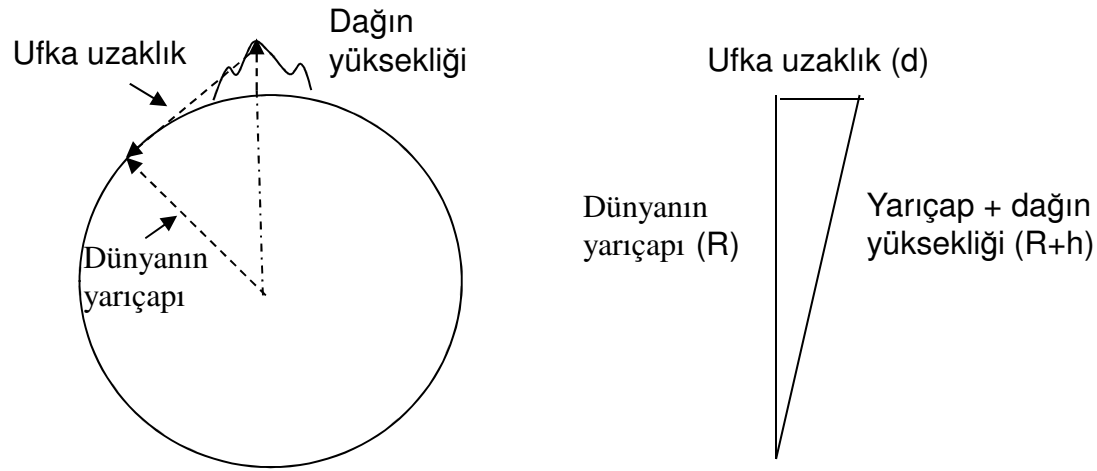
```
Command Window
File Edit Debug Desktop Window Help
>> x=1:5;
>> y=1:3;
>> [new_x,new_y]=meshgrid(x,y)
new_x =
    1     2     3     4     5
    1     2     3     4     5
    1     2     3     4     5
new_y =
    1     1     1     1     1
    2     2     2     2     2
    3     3     3     3     3
```

Diziler aynı boyutta, birbiriyle çarpılabilir.



```
Command Window
File Edit Debug Desktop Window Help
new_x =
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
new_y =
     1     1     1     1     1
     2     2     2     2     2
     3     3     3     3     3
>> A=new_x.*new_y
A =
     1     2     3     4     5
     2     4     6     8    10
     3     6     9    12    15
```

UFKA UZAKLIK



- Ay ve dünya üzerinde bir dağın tepesinden ufka olan uzaklığı bulun.

- Input

- Ayın yarıçapı 1737 km
- Dünyanın yarıçapı 6378 km
- Dağın yüksekliği 0 to 8000m

- Output

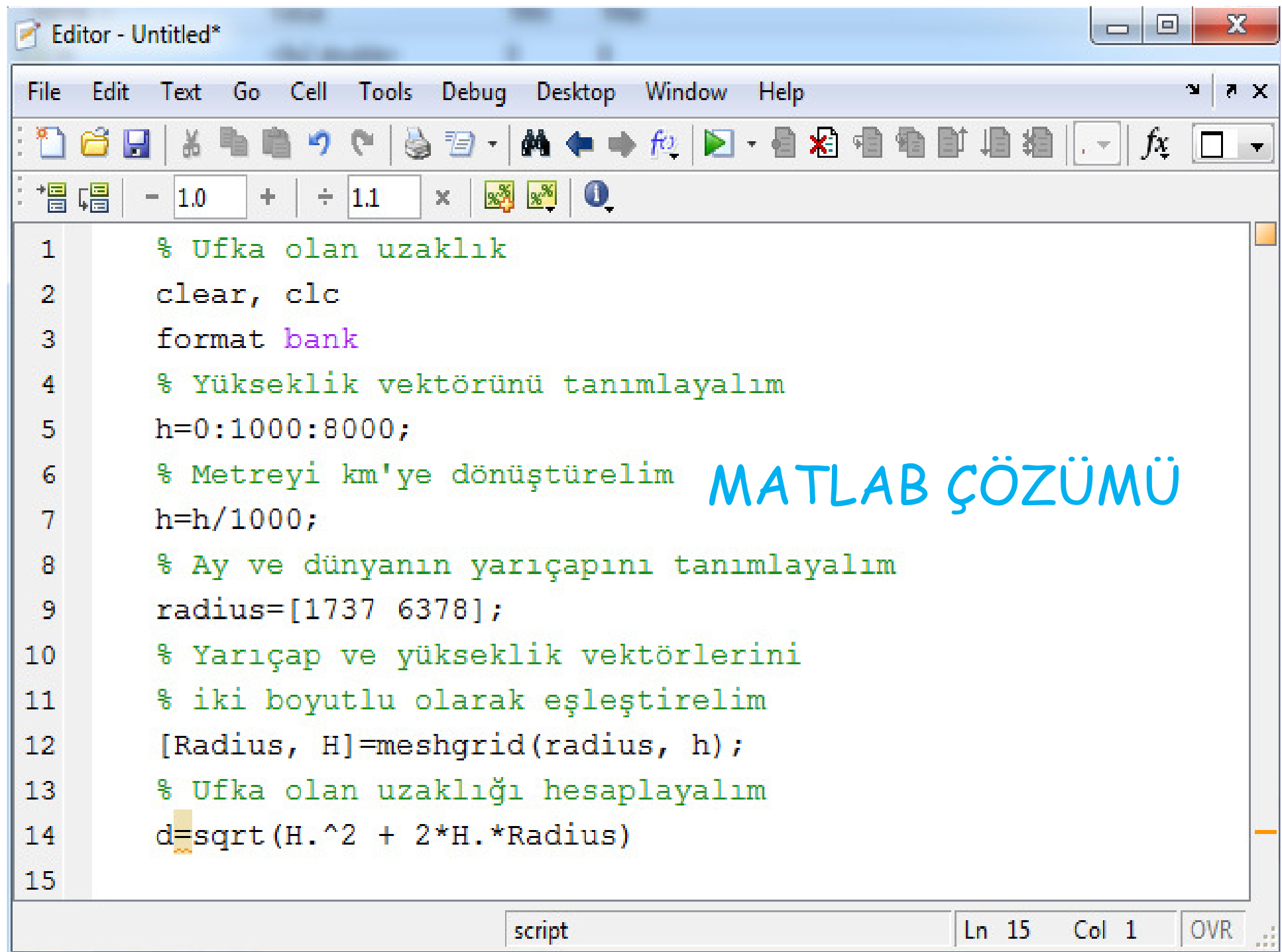
- Km olarak ufka olan uzaklık

$$R^2 + d^2 = (R + h)^2 \quad \text{Pisagor Teoremi}$$

$$d = \sqrt{h^2 + 2Rh}$$

Dünyanın yarıçapı ve 8000m dağ bilgilerini kullanarak:

$$d = \sqrt{(8km)^2 + 2 * 6378km * 8km} = 319km$$



The image shows a MATLAB Editor window titled "Editor - Untitled*". The window has a menu bar with "File", "Edit", "Text", "Go", "Cell", "Tools", "Debug", "Desktop", "Window", and "Help". Below the menu bar is a toolbar with various icons for file operations, editing, and execution. The main editing area contains a MATLAB script with 15 lines of code. The script calculates the distance to the horizon based on height and Earth's radius. The text "MATLAB ÇÖZÜMÜ" is overlaid on the right side of the script. The status bar at the bottom shows "script", "Ln 15", "Col 1", and "OVR".

```
1 % Ufka olan uzaklık
2 clear, clc
3 format bank
4 % Yükseklik vektörünü tanımlayalım
5 h=0:1000:8000;
6 % Metreyi km'ye dönüştürelim
7 h=h/1000;
8 % Ay ve dünyanın yarıçapını tanımlayalım
9 radius=[1737 6378];
10 % Yarıçap ve yükseklik vektörlerini
11 % iki boyutlu olarak eşleştirelim
12 [Radius, H]=meshgrid(radius, h);
13 % Ufka olan uzaklığı hesaplayalım
14 d=sqrt(H.^2 + 2*H.*Radius)
15
```

MATLAB ÇÖZÜMÜ

script Ln 15 Col 1 OVR

M-file çalıştırıldığında aşağıdaki sonuç elde edilir:

```
Command Window

d =

          0          0
      58.95     112.95
      83.38     159.74
     102.13     195.65
     117.95     225.92
     131.89     252.60
     144.50     276.72
     156.10     298.90
     166.90     319.55
```


ÖZEL MATRİSLER

- zeros
 - 0'lerden oluşan bir matris oluşturur
- ones
 - 1'lerden oluşan bir matris oluşturur
- diag
 - Diyagonal elemanları çıkarır ya da birim matris oluşturur
- magic
 - “magic” matris oluşturur

```
Command Window
File Edit Debug Desktop Window Help

>> A=zeros(3)
A =
      0      0      0
      0      0      0
      0      0      0

>> B=ones(3)
B =
      1.00      1.00      1.00
      1.00      1.00      1.00
      1.00      1.00      1.00

>>

Command Window
File Edit Debug Desktop Window Help

>> A=zeros(2,3)
A =
      0      0      0
      0      0      0

>> B=ones(3,2)
B =
      1.00      1.00
      1.00      1.00
      1.00      1.00

>> |
```

```
Command Window
File Edit Debug Desktop Window Help
>> A=[1 2 3; 3 4 5; 1 2 3];
>> diag(A)
ans =
        1.00
        4.00
        3.00
>>
```

diag fonksiyonuna girdi elemanları kare matris olduğunda, diyagonal elemanlar döner.

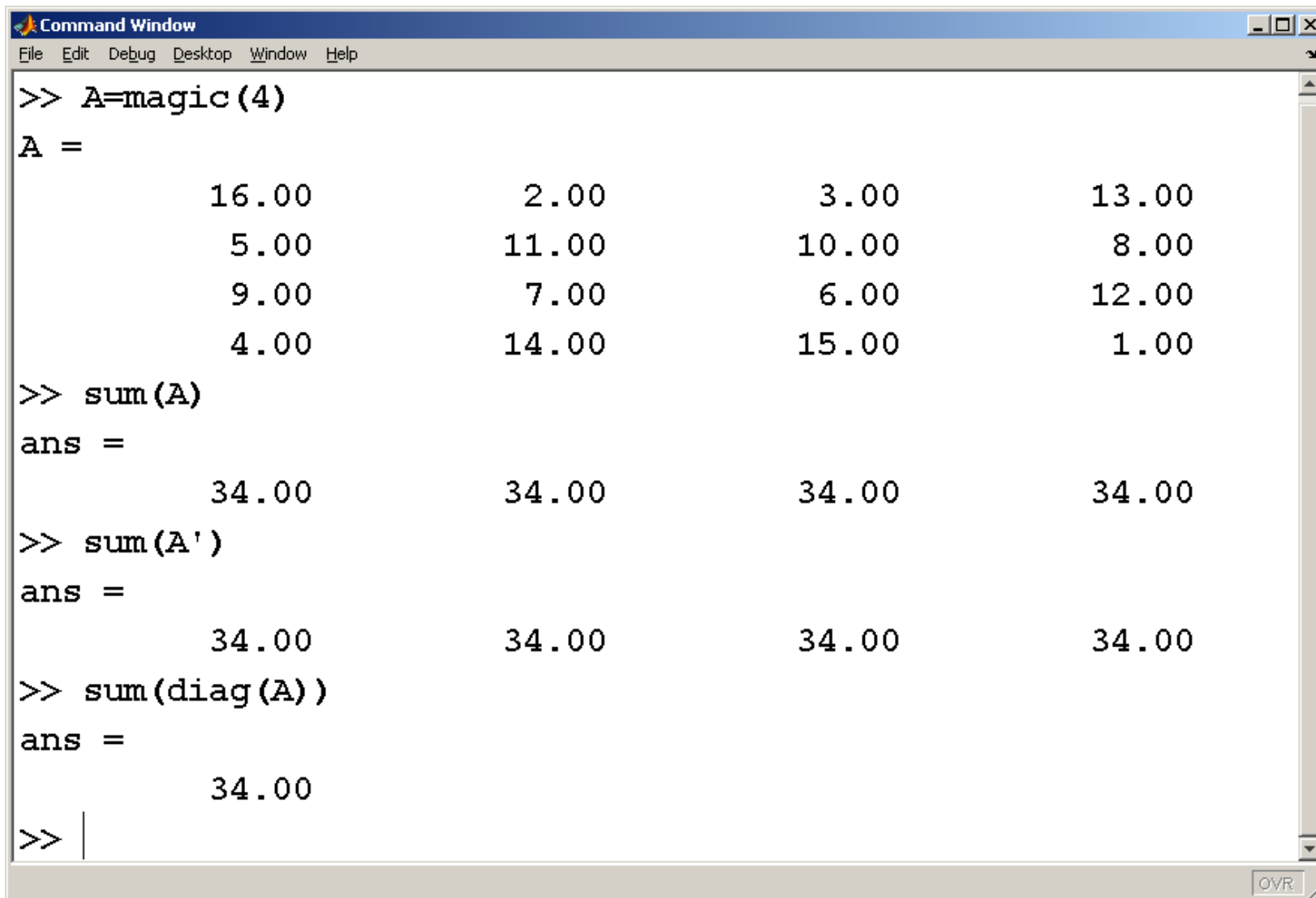
```
Command Window
File Edit Debug Desktop Window Help
>> B=[1 2 3];
>> diag(B)
ans =
        1.00         0         0
         0         2.00         0
         0         0         3.00
>>
```

Girdi vektör olduğunda birim matrisin diyagonalleri olarak kullanılır.

DÜRER'İN SİHİRLİ MATRİSİ

- Satır ve sütun toplamaları eşittir.
>>A = magic(n) : nxn'lik Dürer matrisi oluşturur.
>>sum(A) : A'nın sütun toplamalarını içeren bir satır vektörü oluşturur.
- Matlab genellikle satır yerine sütunlarla çalışır.
- Satır toplamı nasıl hesaplanır?
- Satır toplamalarını hesaplamamanın en kolay yolu transpose alıp tekrar sütun toplamalarını hesaplamaktır. Sonra (gerekirse) sonucun transpose'u alınır.
- Transpose işlemi üstten virgöl (') ile uygulanır.
>>A'
>> sum(A')'

DÜRER'İN SİHIRLİ MATRİSİ



```
Command Window
File Edit Debug Desktop Window Help

>> A=magic(4)
A =
    16.00    2.00    3.00   13.00
     5.00   11.00   10.00    8.00
     9.00    7.00    6.00   12.00
     4.00   14.00   15.00    1.00

>> sum(A)
ans =
    34.00    34.00    34.00    34.00

>> sum(A')
ans =
    34.00    34.00    34.00    34.00

>> sum(diag(A))
ans =
    34.00

>> |
```

The image shows a MATLAB Command Window with the following content:

```
Command Window
File Edit Debug Desktop Window Help

>> A=magic(4)
A =
    16.00    2.00    3.00   13.00
     5.00   11.00   10.00    8.00
     9.00    7.00    6.00   12.00
     4.00   14.00   15.00    1.00

>> sum(A)
ans =
    34.00    34.00    34.00    34.00

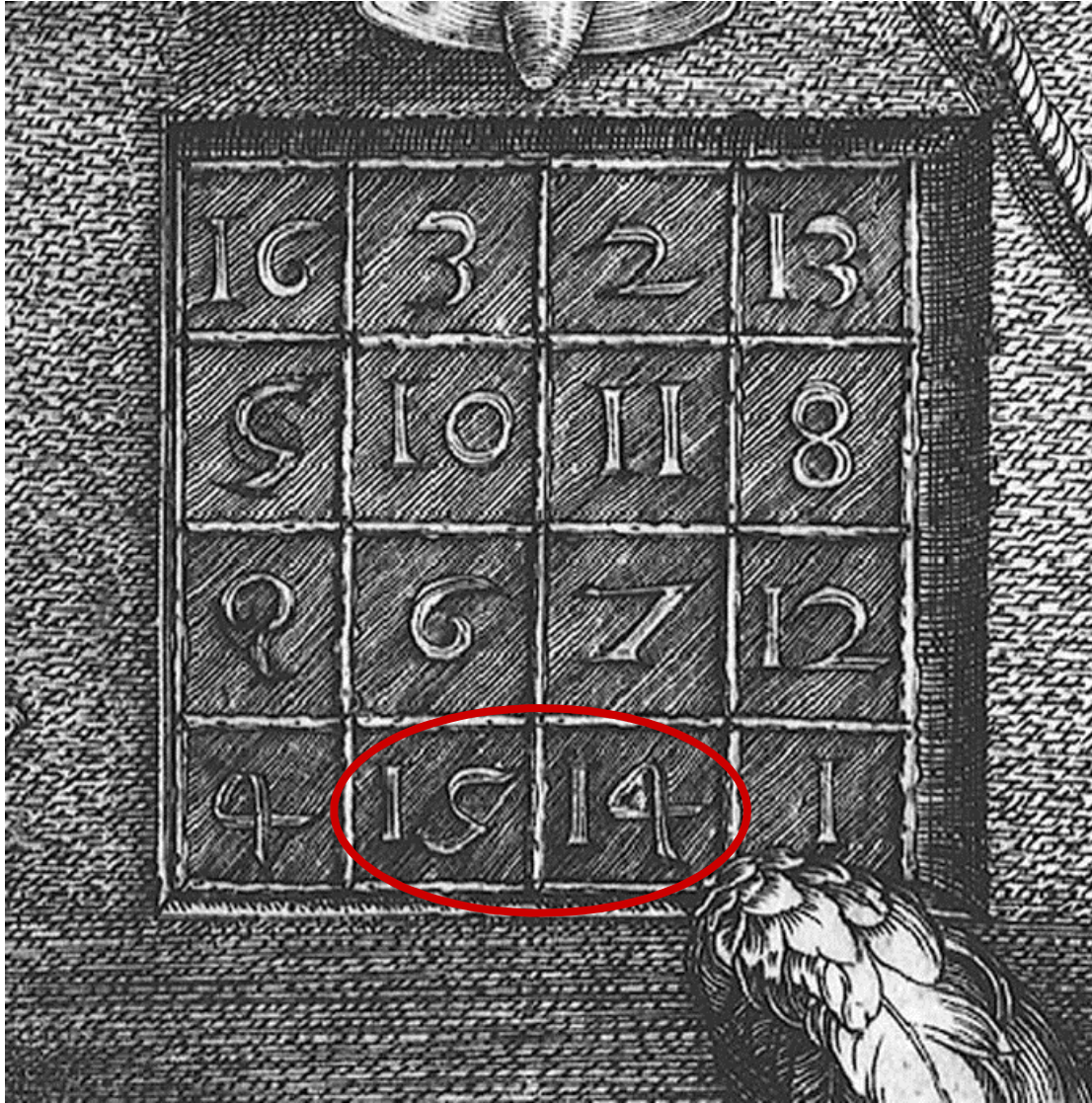
>> sum(A')
ans =
    34.00    34.00    34.00    34.00

>> sum(diag(A))
ans =
    34.00

>> |
```



Melankoli isimli bu ağaç baskı 1514 yılında Albrect Durer tarafından oluşturulmuştur. Meleğin başı üzerinde bir sihirli matris bulunmaktadır.



Albrecht
Dürer bu
sihirli
matrise
tarihi de
eklemiştir.

DURER'İN MATRİSİ MATLAB'İN 4x4 SİHİRLİ MATRİSİNDEN FARKLIDIR

```
>> durer = A(:, [1 3 2 4])
```

```
durer =
```

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Durer tarihi oluşturabilmek için bu iki sütunu yer değiştirmiştir.

```
>> sum(durer)
```

```
ans =
```

34.00	34.00	34.00	34.00
-------	-------	-------	-------

```
>> sum(durer')
```

```
ans =
```

34.00	34.00	34.00	34.00
-------	-------	-------	-------

```
>> sum(diag(durer))
```

```
ans =
```

34.00

ÖRNEK:

1. $a=[2.3 \ 5.8 \ 9]$ matrisini oluşturun.
2. a matrisinin sinüsünü bulun.
3. a 'daki her elemana 3 ekleyin.
4. $b=[5.2 \ 3.14 \ 2]$ matrisini oluşturun.
5. a ve b matrisindeki her elemanı birbiriyle toplayın.
6. a 'nın elemanlarını b 'de karşılık gelen elemanla çarpın.
7. a 'daki her elemanın karesini alın.
8. 0'dan 10'a kadar 1'er artan elemanlardan oluşan bir c matrisi oluşturun.
9. 0'dan 10'a kadar 2'şer artan elemanlardan oluşan bir d matrisi oluşturun.
10. `linspace` fonksiyonunu kullanarak 10'dan 20'ye kadar eşit aralıklı 6 değerden oluşan bir matris oluşturun.

CEVAP:

- `a=[2.3 5.8 9]`
- `sin(a)`
- `a+3`
- `b=[5.2 3.14 2]`
- `a+b`
- `a.*b`
- `a.^2`
- `c=0:10`
- `d=0:2:10`
- `linspace(10,20,6)`

ÖRNEK:

1. -2'den 2'ye 1'er artışı x vektörü oluşturun.
2. Vektörün her elemanının mutlak değerini bulun.
3. Vektörün her elemanının karekökünü bulun.
4. -3 ve +3'ün karekökünü bulun.
 - sqrt fonksiyonunu kullanarak
 - nthroot fonksiyonunu kullanarak
 - $-3^{1/2}$ ve $+3^{1/2}$ 'yi hesaplayarak.
5. -9'dan 12'ye 3'er artışı x vektörünü oluşturun.
 - x'in 2'ye bölünmesinden oluşan sonucu bulun
 - x'in 2'ye bölünmesiyle elde edilen kalanı bulun
6. 5'te elde edilen x vektörü için, e^x değerini bulun, $\ln(x)$ ve $\log_{10}(x)$ değerlerini bulun.
7. x vektöründeki hangi elemanların pozitif olduğunu sign fonksiyonunu kullanarak bulun.
8. Formatı rat'a dönüştürün ve x vektörünü 2'ye bölün

CEVAP

1. `x= -2:2;`
2. `abs(x)`
3. `sqrt(x)`
4. `sqrt(-3)`
 - `sqrt(3)`
 - `nthroot(-3,2)` % x negatifse n tek sayı olmalı aksi halde hata verir
 - `nthroot(3,2)`
 - `(-3)^(1/2)`
 - `3^(1/2)`
5. `x=-9:3:12`
 - `x/2`
 - `rem(x,2)`
6. `exp(x)`
 - `log(x)`
 - `log10(x)`
7. `sign(x)`
8. `format rat`
 - `x/2`
 - `format short` % default formata döner