

# MATLAB

The Language of Technical Computing

0.8147  
0.9058  
0.1270  
0.9134  
0.6324

0.0975  
0.2785  
0.5469  
0.9575  
0.9649

0.1576  
0.9708  
0.9572  
0.4854  
0.8003

```
load('index.txt')  
= diff  
= map  
= get(gca, 'Position');  
1:12  
xpos = (rex(n-1,4)/4)*ax(3)+ax(4);  
Pos = ((2-floor(n-1)/3)*ax(1)+ax(2),
```

## MATLAB'E GİRİŞ



- **MATLAB® (MATRIX LABORATORY) SAYISAL HESAPLAMA DİLİDİR. MATHWORKS FİRMASI TARAFINDAN GELİŞTİRİLMİŞ OLUP, MATRİS İŞLENMESİNE, FONKSİYONLAR VE VERİ ÇİZİLMESİNE, ALGORİTMALAR UYGULANMASINA, KULLANICI ARAYÜZÜ OLUŞTURULMASINA VE DİĞER DİLLERLE YAZILMIŞ PROGRAMLAR İLE ETKİLEŞİM OLUŞTURULMASINA İZİN VERMEKTEDİR.**
- **(MATLAB' DE YAZILAN BİR PROGRAM C, C++, JAVA, VE FORTRAN DİLLERİNİ ÇEVİRİLEBİLMEKTEDİR.**



1970'lerin sonunda Cleve Moler tarafından yazılan Matlab programının tipik kullanım alanları:

- Matematiksel (nümerik ve sembolik) hesaplama işlemleri
- Algoritma geliştirme ve kod yazma yani programlama
- Lineer cebir, istatistik, Fourier analizi, filtreleme, optimizasyon, sayısal integrasyon vb. konularda matematik fonksiyonlar
- 2D ve 3D grafiklerinin çizimi
- Modelleme ve simülasyon (benzetim)
- Grafikselsel arayüz oluşturma
- Veri analizi ve kontrolü
- Gerçek dünya şartlarında uygulama geliştirme şeklinde özetlenebilir.



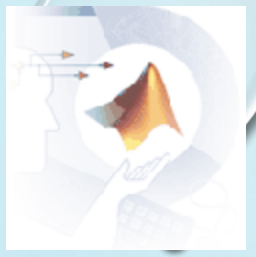
<http://www.mathworks.com/matlabcentral/>

**MATLAB**, yüksek performanslı bir uygulama yazılımı ve bir programlama dilidir.

MATLAB'in temelindeki yapı, boyutlandırma gerektirmeyen matrislerdir.

Yaptığımız tüm girdi ve çıktılar, belirteç gerektirmeksizin bir matris tanımlar.

İlk olarak Fortran dili ile yazılan MATLAB, daha sonra C ile yazılmıştır.

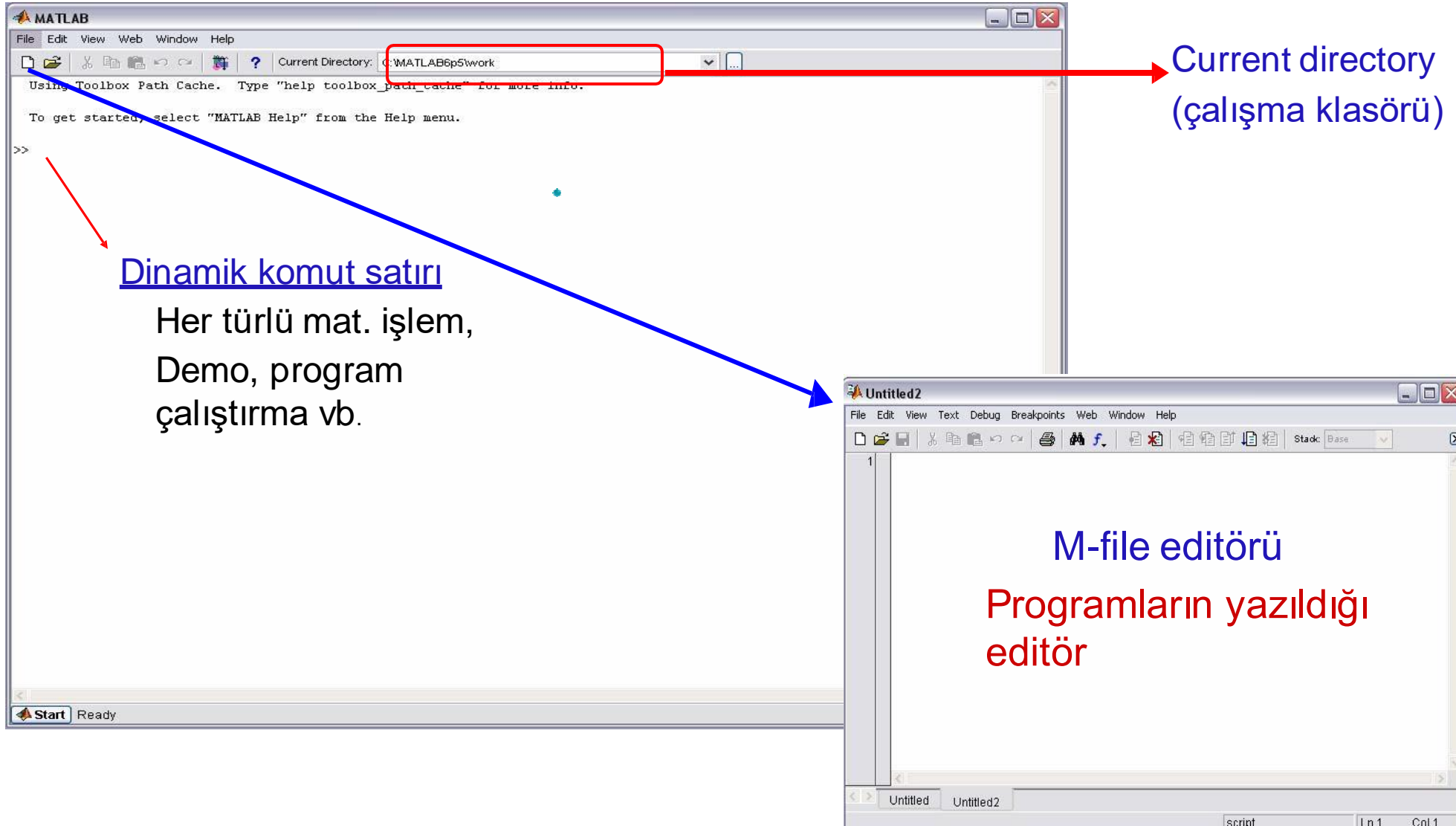


Matlab'de hazır programlar vardır. Bu programlara **fonksiyon** adı verilir.

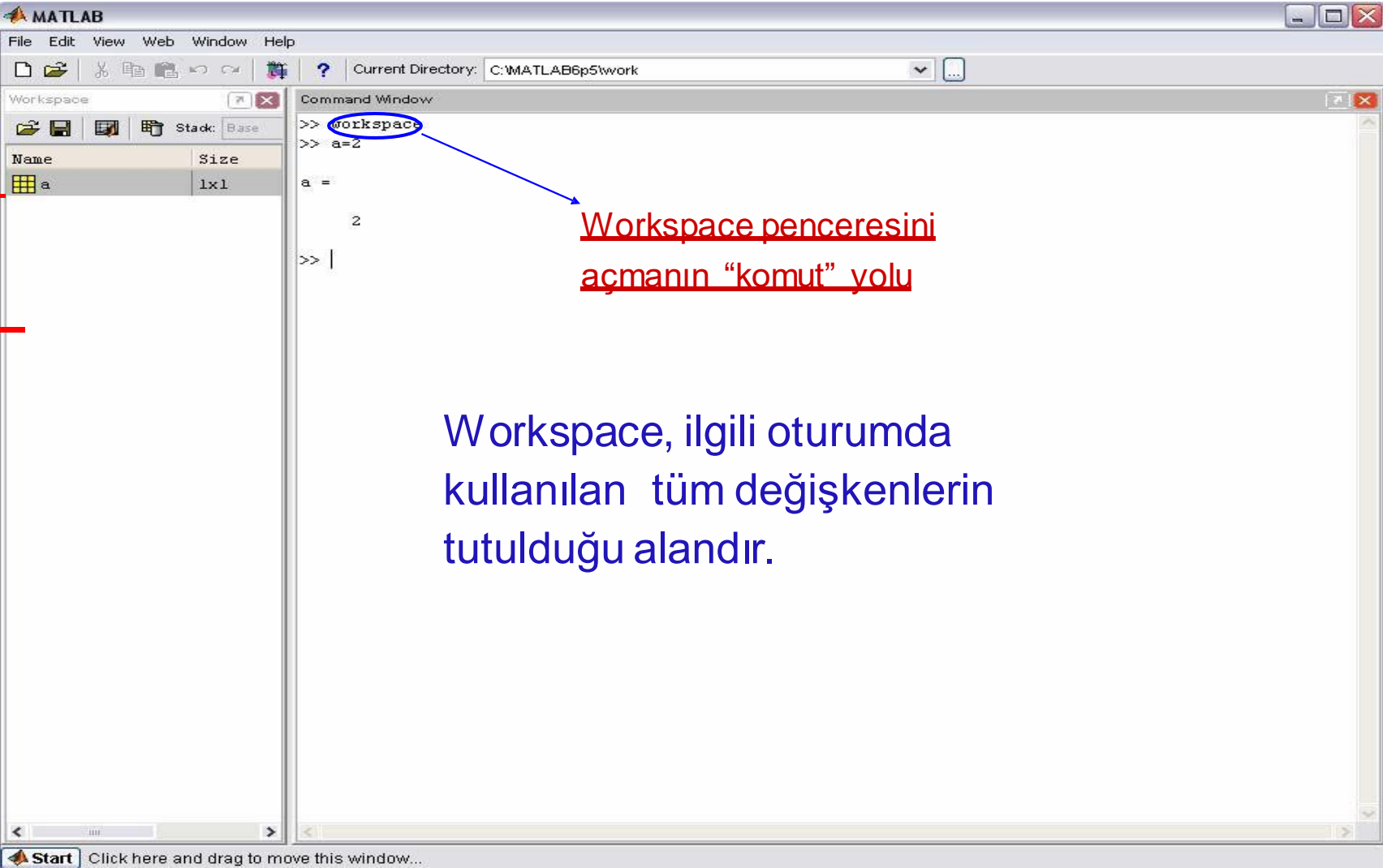
Matlab fonksiyonlarının kullanımı, matematikteki  $y=f(x)$  fonksiyonunun kullanımıyla özdeştir.

Örneğin,  $a=\sin(x)$  fonksiyonunda, **sin** fonksiyonu,  $x$  açısının (input- girdi) değerini hesaplar; kullanıcı bu değeri, örneğin, bir  $a$  değişkenine atar.  $a$  değeri **sin** fonksiyonunun bir çıktısıdır (output).

# MATLAB/Command window (komut penceresi)



## MATLAB/Workspace (İş alanı)



The image shows the MATLAB interface with the Workspace and Command Window panes. The Workspace pane on the left displays a table with the following content:

Name	Size
a	1x1

The Command Window on the right shows the following commands and output:

```
>> workspace  
>> a=2  
  
a =  
  
    2  
>> |
```

Annotations in the image include:

- A blue arrow pointing from the text "Workspace penceresini açmanın 'komut' yolu" to the word "workspace" in the Command Window.
- A red arrow pointing from the text "Atanan değişken" to the variable "a" in the Workspace pane.
- A red arrow pointing from the text "Workspace penceresi" to the Workspace pane.

Workspace, ilgili oturumda kullanılan tüm değişkenlerin tutulduğu alandır.



## **MATLAB KULLANIMINDA TEMEL KURALLAR:**

- PROGRAMIN DİLİ VE YARDIM BİLGİLERİ TAMAMEN İNGİLİZCEDİR.
- KOMUT TEMELLİ yazılımdır.
- KÜÇÜK-BÜYÜK HARF AYRIMI VARDIR. ToT VE TOT FARKLI ALGILANIR.
- » İŞARETİ KOMUT PROMPT'UDUR.
- KOMUTLAR *ENTER* İLE YÜRÜTÜLÜR.



## **clc VE clear KOMUTLARI**

**clc** komutu komut penceresini temizler.

**clear** komutu bütün değişkenleri ve fonksiyonları bellekten siler. (Bütün değişkenleri çalışma alanından (workspace) çıkarır.)

**Lütfen Aşağıdaki Komutları Komut Penceresinden Çalıştırın:**

```
>>help clc
```

```
>>help clear
```

```
>>help abs
```

```
>>help sqrt
```

**Daha detaylı yardım için (komutun kullanımına ait örnekleri de görmek için) **doc** komutundan faydalanabilirsiniz.**

```
>>doc sqrt
```

# MATLAB'DE DEYİMLER(ifadeler) VE BİLEŞENLERİ

**MATLAB** komut penceresinden doğrudan çalıştırılabilen ya da bir **MATLAB** dosyası içerisine kaydedilebilen ve **MATLAB** tarafından yorumlanabilen her şey **MATLAB** deyimidir. **MATLAB** deyimleri *değişkenler*, *rakamlar*, *operatörler* ve *fonksiyonlar* kullanılarak oluşturulurlar.

**Değişkenler:** Deyimler içerisinde sayısal değerlerin yerini alan ifadelerdir. **MATLAB** bir değişken ile karşı karşıya geldiğinde, otomatik olarak bu değişken oluşturulur ve yeteri kadar bellek ayrılır. Eğer değişken daha önceden tanımlı ise **MATLAB** onun içeriğini değiştirir ve gerekirse yeni bellek ayırır.

Örneğin, `>> x=50`

komut satırından işletildiğinde **MATLAB** “x” adında bir değişken oluşturur ve 50 değerini bu değişkene atar. Diğer bilgisayar dillerinde olduğu gibi **MATLAB**'in de değişken isimleri konusunda bazı kuralları vardır.



The screenshot shows the MATLAB Command Window and Workspace. In the Command Window, the command `>> x=50` has been entered and executed. The output shows `x =` followed by `50`. In the Workspace, a variable named `x` is listed with a value of `50` and a minimum value of `50`.

Name	Value	Min
x	50	50



**İFADELER :** Matematiksel ve metinsel gösterimler ile işlemler ifadelerle sağlanır.

- Sayılar (Numbers)
- Değişkenler (Variables)
- İşleçler (Operators)
- Fonksiyonlar (Functions)

➤ Matlab'de ifadeler genellikle **değişken = ifade** veya sadece **ifade** formundadır.

➤ Örnek olarak

```
>> x=4*sqrt(5)
```

```
x =
```

```
8.9443
```

ifadesinde x değişken , 4 sayı , \* işleç , sqrt fonksiyondur.

➤ Bir ifade = işareti ile bir değişkene atanmamış ise Matlab otomatik olarak sonucu, **ans** (answer) adı verilen özel bir değişkende saklar.



- Bir ifadenin sonuna ; işareti eklediğinizde ekrana yazılmaz.
- Birden fazla ifade tek satırda aralarına , ya da ; koyarak yazılabilir. Yazımda boşluk sayısı etkisizdir.
- Sadece ilgili değişken adını yazarak o değişkeni çağırabilirsiniz.

## Değişken Tanımlama Kuralları

1. Değişken isimleri daima bir harf ile başlamalıdır. İlk harfi başka harfler, rakamlar ve alt çizgiler ( \_ ) izleyebilir. Örneğin **1sayi** uygun bir değişken ismi değilken, **sayi\_1** kullanılabilir bir değişken adıdır.

```
Command Window
>> 1sayi
    1sayi
    |
Error: Unexpected MATLAB expression.

fx >> |
```

2. Değişken isimleri noktalama işaretleri ve boşluk(lar) *barındırmamalıdır*. **sayi.1** veya **sayi 2** (2 den önceki boşluğa dikkat) şeklinde tanımlanan değişken adları bu kural ihlaline 2 güzel örnektir.

```
Command Window
>> sayi.1
    sayi.1
    |
Error: Unexpected MATLAB expression.

>> sayi 1
Undefined function 'sayi' for input arguments of type 'char'.

fx >> |
```

## Değişken Tanımlama Kuralları

3. Değişken isimleri içerisinde Türkçemize özgü küçük veya büyük “ç, ğ, ı, ö, ş, ü” karakterleri kullanılmamalıdır.

```
Command Window
>> sayı
    sayı
    |
Error: The input character is not valid in MATLAB statements or expressions.
fx >> |
```

4. Değişken isimlerinin küçük ve büyük harf kullanımına duyarlı olduğu unutulmamalıdır. **orta**, **Orta**, **ORTA** veya **orTa** adlı değişkenler **MATLAB** için farklı değişkenlerdir.

```
Command Window
>> orta=3;
>> Orta=10000;
>> orta

orta =

     3

>> Orta

Orta =

    10000
```

5. Değişken isimleri en fazla 63 karakter içerebilir ve bundan fazlası **MATLAB** tarafından dikkate alınmaz.

# MATLAB Programlama Ortamının Tanıtımı

**Komut Penceresi:** **MATLAB** ile iletişim kurulan ana penceredir. **MATLAB** yorumlayıcısı, kullanıcıdan gelecek komutları kabul etmeye hazır olduğunu gösteren “ **>>** ” biçiminde bir simge görüntüler. Örneğin, **4\*25+6\*52+2\*99** gibi basit bir matematiksel işlemi yapmak için

**>> 4\*25+6\*52+2\*99** ifadesini yazıp **ENTER** tuşuna basarız.

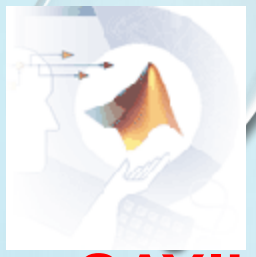
**ans=**  
**610**

**Komut Satırında Yanlışların Düzeltilmesi:** Klavyede yer alan ok tuşları komut satırında yapılan yanlışlıkların düzeltilmesine imkan verir. Bunlar yukarı “**↑**” aşağı “**↓**” sol “**←**” sağ “**→**”. Yukarı tuşu yardımıyla bir önceki satır tekrar görüntülenerek sağ ve sol tuşları ile yanlış yazılı yere imleç taşınarak düzenleme gerçekleştirilir.

**Sonucun Ekranda Görüntülenmesini Gizleme:** Bir deyim yazıp, **ENTER** tuşuna basınca sonuçlar ekranda otomatik olarak görüntülenir. Buna karşılık, deyim sonuna “**;**” ilave edilecek olursa yapılan hesaplamalar ekranda görüntülenmez.



```
Command Window
>> 4*25+6*52+2*99
ans =
    610
>> 4*25+6*52+2*99;
fx >> |
```



## SAYILAR

- Ondalık sayılar Türkçemizdeki 3,5 yerine 3.5 şeklinde gösterilir (0.0001 veya 0.65 yerine .65).
- Bilimsel notasyon gösterimi olan **e** (veya E) harfi 10'nun kuvvetini temsil eder.
  - $2e4=2.10^4=2000$  veya  $1.65e-20=1.65.10^{-20}$  demektir.
- Kompleks sayılarda imajiner (sanal) kısımlar **i** veya **j** ekini alır. **i** veya **j** ile gösterimlerinde bir fark yoktur.
  - $(1+3i$  veya  $1+3*i$  veya  $1+i*3$  ama  $1+i3)$  /  $1+\text{sqrt}(3)*j$  /  $-5i$
- Matlab'de tüm sayılar, yaklaşık  $2.10^{308}$  ve  $2.10^{-308}$  arasında değişir.



## Aritmetik Operatörler

**MATLAB**, deyimler içerisinde aşağıdaki aritmetik operatörleri ve öncelik kurallarını kullanır.

İşlem	Operatör	Örnek
Toplama, $a+b$	+	$2+3$
Çıkarma, $a-b$	-	$5-2$
Çarpma, $a*b$	*	$3*4$
Bölme, $a/b$	/	$15/7$
Üs alma, $a^b$	^	$2^3$
Parantez $a*(b+c)$	( )	$2*(3+5)$

**Aritmetik Operatörlerin Öncelik Sıraları:** Tek bir matematiksel deyim içinde birden fazla işlem bir arada bulunabildiğine göre hangi işlemin öncelik hakkına sahip olduğunun bilinmesi yerinde olacaktır. **MATLAB**'de işlemlerin öncelik listesi aşağıda verilmiştir.

### Öncelik

- 1.
- 2.
- 3.
- 4.

### İşlem

Parantez (İçten Dışa Doğru)

Üs Alma (Soldan Sağa Doğru) ( $2^2^3=???$ )

Çarpma ve Bölme (Soldan Sağa Doğru)

Toplama ve Çıkarma (Soldan Sağa Doğru)

# Uygulamalar

## Uygulama\_1 :

Matematiksel Yazılım	MATLAB de Yazılım
$ab-c+d-6+da$	$a*b-c+d-6+d*a$

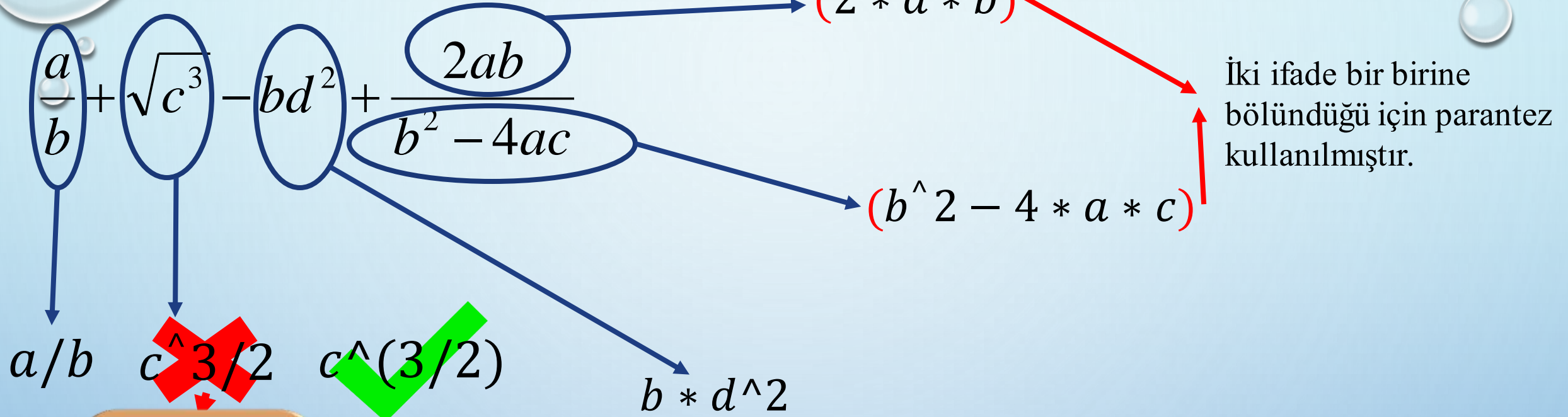
```
>> a=1;b=2;c=3;d=4;e=5;f=6;  
>> a*b-c+d-6+d*a  
  
ans =  
  
1
```

## Uygulama\_2 :

Matematiksel Yazılım	MATLAB de Yazılım
$b+c^3-d/8-b^2c$	$b+c^3-d/8-b^2*c$

```
>> b+c^3-d/8-b^2*c  
  
ans =  
  
16.5000
```

### Uygulama\_3 :



```
>> 4^3/2
ans =
    32

>> 4^(3/2)
ans =
    8
```

$$a/b + c^{(3/2)} - b*d^2 + (2*a*b)/(b^2 - 4*a*c)$$

**a=1, b=2, c=3, d=4**

```
>> a/b+c^(3/2)-b*d^2+(2*a*b)/(b^2-4*a*c)
ans =
   -26.8038
```

# MATLAB/TEMEL KOMUTLAR

<code>clc</code>	Command window'u temizler.
<code>clear</code>	İlgili oturumda atanmış tüm değişkenleri siler.
<code>clear a</code>	Yalnızca "a" değişkenini siler.
<code>demo</code>	Matlab demosunu çalıştırır.
<code>date</code>	Gün-Ay-Yıl'ı görüntüler (Örneğin, 17-Oct-2009)
<code>exit</code>	Matlab oturumundan çıkar.
<code>help</code>	Yardım menüsünü açar.
<code>help f_na</code>	f_na fonksiyonu hakkında bilgi verir.
<code>save d a</code>	a değişkenini d dosya ismiyle <b>mat</b> uzantılı olarak kaydeder.
<code>load d</code>	a değişkenini d dosyasından geri çağırır.

**Save ve load komutları, matris vb. yapıların kaydedilmesi için çok önemlidir.**

# MATLAB/TEMEL DOSYA TÜRRLERİ

- \* **.m** MATLAB program dosyaları
- \* **.fig** Grafik dosyaları ve GUI'lerin grafik parçaları
- \* **.mat** Değişken ve matris dosyaları
- \* **.p** pre-parsed pseudo-code dosyaları (bu dosyaların içeriği görüntülenemez ancak program olarak çağrılabilir, yani MATLAB'de çalıştırılabilir!)

# MATLAB / Matrislerin Girilmesi

matris ve vektörler [ ] köşeli parantezleri ile tanımlanır.

Matris ve vektör girmenin 3 farklı yolu vardır:

Örneğin:

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 11 \\ 100 & 1 & 4 \end{bmatrix}$$

1.yol

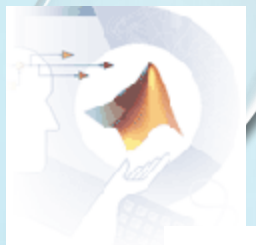
```
A=[1 3 5  
7 8 11  
100 1 4]
```

2.yol

```
A=[1 3 5;7 8  
11;100 1 4]
```

3.yol

```
A(1,1)=1,    A(1,2)=3,    A(1,3)=5  
A(2,1)=7,    A(2,2)=8,    A(2,3)=11  
A(3,1)=100,  A(3,2)=1,    A(3,3)=4
```



$$M = \begin{bmatrix} 4 & -15 & 3 & 27 \\ 1 & 9 & 16 & -7 \\ 123 & 25 & 34 & 2 \end{bmatrix}$$

Command Window

```
>> satirVektoru=[4 -15 3 27]
```

```
satirVektoru =
```

```
4 -15 3 27
```

```
>> satirVektoru=[4, -15, 3, 27]
```

```
satirVektoru =
```

```
4 -15 3 27
```

$$\text{sutunVektoru} = \begin{bmatrix} 4 \\ 1 \\ 123 \end{bmatrix}$$

Command Window

```
>> sutunVektoru=[4;1;123]
```

```
sutunVektoru =
```

```
4
```

```
1
```

```
123
```

## Karakter Dizilerinin Değişkenlere Değer Olarak Atanması

Command Window

```
>> isim='deniz'
```

```
isim =
```

```
deniz
```

```
>> isim=['d' 'e' 'n' 'i' 'z']
```

```
isim =
```

```
deniz
```



Matris ve vektörler \*.mat uzantılı olarak **save** komutuyla kaydedilir, **load** ile de istenilen yerden geri çağrılır.

Örneğin, girilmiş bir **a** matrisini “**D:\gs**” klasörüne “**katsayilar.mat**” olarak kaydetmek isteyelim: Bunun için aşağıdaki komut dizisi kullanılır;

```
save D:\gs\katsayilar a
```

katsayilar.mat olarak kaydedilen a matrisinin herhangi bir zamanda geri çağırılması için,

```
load D:\gs\katsayilar
```

komut dizisi kullanılır. Geri çağırma işleminden sonra, ilgili matris a dizisi olarak workspace’de kaydedilir (workspace’e kaydetme işleminin geçici olduğunu hatırlayınız!)

Yeni bir matrisi katsayilar.mat olarak kaydettiğimizde, önceki matrisi bir daha **görme imkanı** kalmaz. Yani **save overwrite (üzerine yazma)** özelliğlidir.



# MATLAB/SAYI FORMATLARI

```
>> format long
>> a=1.123000123123123123;
>> a
a =
1.12300012312312 16 karakter
```

```
>> a=100004545.99923423499111;
>> a
a =
1.000045459992342e+008 108
```

```
>> format short
>> a=1.123000123123123123;
>> a
a =
1.1230 6 karakter
```

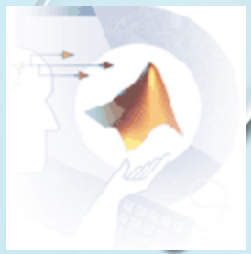
```
>> a=100004545.99923423499111;
>> a
a =
1.0000e+008 108
```

Bir sayının istenilen hanesinin **gösterilmesi** için `fprintf` veya `sprintf` komutları kullanılır:

```
>>a=100004545.99923423499111;
>>fprintf( '%1.10f' ,a)

100004545.9992342300
```

Matris elemanlarının istenilen hanede gösterilmesi için, `printmatrix` fonksiyonu oldukça kullanışlıdır (File-exchange sayfasından download edilebilir!)



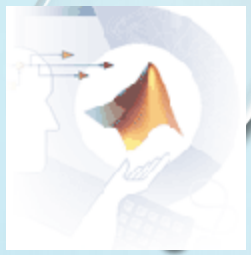
# MATLAB / Temel lineer cebir komutları

- **trace (a)** Bir a matrisinin izini (köşegen elemanlarının toplamını) hesaplar.
- **diag (a)** Bir kare a matrisinin köşegen elemanlarını bir sütun vektöre atar. Ya da a bir vektör ise köşegenleri bu vektörün elemanlarından oluşan bir köşegen matris oluşturur.
- **sum (a)** a matrisinin her bir sütununun toplamını hesaplar. a bir vektör ise sonuç, vektör elemanlarının toplamı olur.
- **triu (a)** Bir matrisin üst üçgen matrisini oluşturur.
- **tril (a)** Bir matrisin alt üçgen matrisini oluşturur.
- **zeros (m, n)**  $m \times n$  boyutlu sıfır matrisi oluşturur.
- **ones (m, n)**  $m \times n$  boyutlu elemanları "1" olan matris oluşturur.
- **eye (m)**  $m \times m$  boyutlu birim matris oluşturur.



# MATLAB / Temel lineer cebir komutları

- **inv(a)** Bir a kare matrisinin tersini (inversini) alır.
- **a'** a matrisinin devriğini (transpozmesini) alır.
- **det(a)** a matrisinin determinantını hesaplar.
- **a+b** Boyutları aynı olan a ve b matrisini toplar.
- **a-b** Boyutları aynı olan a ve b matrislerinin farkını alır.
- **a\*b** Sütun sayısı m olan a matrisiyle satır sayısı m olan b matrisini çarpar.
- **a/b** b düzenli kare bir matrisse (determinantı sıfırdan farklıysa), aynı boyutlu a matrisiyle;  
**a\*inv(b)** işlemini yapar.
- **a.\*b** Boyutları aynı olan a ve b matrislerinin elemanların karşılıklı olarak çarpar.
- **a./b** Boyutları aynı olan a ve b matrislerinin elemanlarını karşılıklı oranlar



# « MATLAB / Temel matris operatörleri

- $a(:)$  a matrisinin sütunlarının ardarda dizilmesinden oluşan bir sütun vektör oluşturur (vec operatörü)
- $a(:, i)$  a matrisinin i. sütununu alır.
- $a(j, :)$  a matrisinin j. satırını alır.
- $a(:, [i \ j])$  a matrisinin i ve j. sütununu alır.
- $a([i \ j], :)$  a matrisinin i ve j. satırını alır.
- $e=a:b:n$  a, (a+b), ..., n sayılarından oluşan bir satır vektör oluşturur.

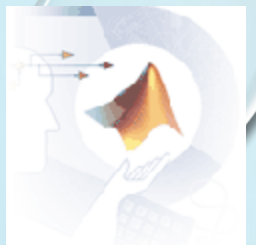
Örneğin,

- $e=1:1:n$ , 1 ile n arasındaki tam sayılardan oluşan bir vektör.
- $e=2:2:n$ , 1 ile n arasındaki çift sayılardan oluşan bir vektör.
- $e=1:2:n$ , 1 ile n arasındaki tek sayılardan oluşan bir vektör.
- $e=-10:0.1:n$ , -10'dan 0.1 artımla n'ye kadar olan sayılardan oluşan bir vektör.



# MATLAB / Temel matris operatörleri

- **length (a)** a matrisinin sütun sayısını verir. a bir vektör ise sonuç, a vektörünün eleman sayısıdır.
- **[m,n]=size (a)** a matrisinin satır sayısını(m) ve sütun sayısını(n) verir.
- **max (a)** Bir a vektörünün en büyük elemanını gösterir.
- **min (a)** Bir a vektörünün en küçük elemanını gösterir.
- **[m,i]=max (a)** Bir a sütun vektörünün en büyük elemanını(m) ve bunun satır numarasını verir.
- **[m,i]=min (a)** Bir a sütun vektörünün en küçük elemanını(m) ve bunun satır numarasını verir.
- **sort (a)** Bir a vektörünün elemanlarını küçükten büyüğe sıralar.
- **a (:,i)=[ ]** A'nın i. sütununu siler.
- **a (i,:)=[ ]** A'nın i. Satırını siler.


$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 11 \\ 100 & 1 & 4 \end{bmatrix}$$

- 1) A matrisini giriniz.
- 2) A matrisinin determinantını hesaplayınız.
- 3) A matrisinin tersini bulunuz. Çıkan sonucu bir B matrisine atayınız.
- 4)  $A*B$  işlemini yapınız. Elde edilen sonucu irdeleyiniz.
- 5) A matrisinin 1. sütununu  $a_1$ , 3. sütununu  $a_3$  vektörlerine atayınız.
- 6) Köşegenleri A matrisinin köşegenlerinden oluşan bir C köşegen matrisi oluşturunuz.
- 7)  $a_1$ 'in devriği ile  $a_3$  vektörünü çarpınız.
- 8)  $a_1$  ile  $a_3$  vektör elemanlarını karşılıklı çarpınız.
- 9) A'nın 3. satırını, diğer satır elemanlarını girmeden,  $[5 \ 6 \ 7]$  olarak değiştiriniz.
- 10) A'nın 1 ve 2. satırlarını siliniz.

```
1 >> A=[1 3 5;7 8 11;100 1 4];
2 >> det(A)
ans =
    -728
3 >> inv(A)
ans =

    -0.0288    0.0096    0.0096
   -1.4725    0.6813   -0.0330
    1.0893   -0.4107    0.0179
>> B=ans;
4 >> A*B
ans =
    1.0000         0    0.0000
         0    1.0000    0.0000
         0    0.0000    1.0000
    Birim matris
5 >> a1=A(:,1);a3=A(:,3);
6 >> C=diag(diag(A));
7 >> a1'*a3
ans =
    482
```

```
8 >> a1.*a3
ans =
     5
    77
   400
9 >>A(3,:)= [5 6 7]
A =

     1     3     5
     7     8    11
     5     6     7
10 >> A([1 2],:)=[]
A =

     5     6     7
>>
```

# Matlab' da Trigonometrik Fonksiyonlar

Fonksiyon	İşlev	Örnek kullanım
sin(x)	Radyan cinsinden girilen açının sinüsünü verir.	<pre>&gt;&gt; sin(pi/2) ans=      1  &gt;&gt; sin([0 pi/2 pi 3*pi/2]) ans=      0     1.0000     0.0000    -1.0000  &gt;&gt; sin([0 pi/2; pi 3*pi/2]) ans=      0     1.0000     0.0000    -1.0000</pre> <p><b>Not:</b> Tablolarda görülen tüm fonksiyonlarda girdi olarak yukarıdaki gibi tek sayı, dizi ve matris kullanılabilir.</p>
sind(x)	Derece cinsinden girilen açının sinüsünü verir.	<pre>&gt;&gt; sind(90) ans=      1</pre>
asin(x)	Sinüsü girilen açıyı radyan cinsinden verir.	<pre>&gt;&gt; asin(1) ans=     1.5708</pre>

## pi Sabiti

Command Window

```
>> pi
ans =
    3.1416

>> r=3
r =
     3

>> alan=pi*r^2
alan =
    28.2743

>> cevre=2*pi*r
cevre =
    18.8496
```



# Matlab' da Trigonometrik Fonksiyonlar

Fonksiyon	İşlev	Örnek kullanım
<b>asind(x)</b>	Sinüsü girilen açıyı derece cinsinden verir.	<pre>&gt;&gt; <b>asind</b>(1) ans=     90</pre>
<b>cos(x)</b>	Radyan cinsinden girilen açının kosinüsünü verir.	<pre>&gt;&gt; <b>cos</b>(0) ans=     1</pre>
<b>cosd(x)</b>	Derece cinsinden girilen açının kosinüsünü verir.	<pre>&gt;&gt; <b>cosd</b>(0) ans=     1</pre>
<b>acos(x)</b>	Kosinüsü girilen açıyı radyan cinsinden verir.	<pre>&gt;&gt; <b>acos</b>(1) ans=     0</pre>
<b>acosd(x)</b>	Kosinüsü girilen açıyı derece cinsinden verir.	<pre>&gt;&gt; <b>acosd</b>(1) ans=     0</pre>
<b>tan(x)</b>	Radyan cinsinden girilen açının tanjantını verir.	<pre>&gt;&gt; <b>tan</b>(pi/4) ans=     1</pre>
<b>tand(x)</b>	Derece cinsinden girilen açının tanjantını verir.	<pre>&gt;&gt; <b>tand</b>(45) ans=     1</pre>

# Matlab' da Trigonometrik Fonksiyonlar

Fonksiyon	İşlev	Örnek kullanım
<b>atan(x)</b>	Tanjantı girilen açıyı radyan cinsinden verir.	<pre>&gt;&gt; atan(1) ans =     0.7854</pre>
<b>atand(x)</b>	Tanjantı girilen açıyı derece cinsinden verir.	<pre>&gt;&gt; atand(1) ans =     45</pre>
<b>cot(x)</b>	Radyan cinsinden girilen açının kotanjantını verir.	<pre>&gt;&gt; cot(pi/4) ans =     1</pre>
<b>cotd(x)</b>	Derece cinsinden girilen açının kotanjantını verir.	<pre>&gt;&gt; cotd(45) ans =     1</pre>
<b>acot(x)</b>	Kotanjantı girilen açıyı radyan cinsinden verir.	<pre>&gt;&gt; acot(1) ans =     0.7854</pre>
<b>acotd(x)</b>	Kotanjantı girilen açıyı derece cinsinden verir.	<pre>&gt;&gt; acotd(1) ans =     45</pre>
<b>hypot(x)</b>	Dik kenarlar verildiğinde hipotenüsü hesaplar.	<pre>&gt;&gt; hypot(3,4) ans =     5</pre>

# Matlab' de logaritmik ve Üstel Fonksiyonlar

Fonksiyon	İşlev	Örnek kullanım
<b>exp(x)</b>	$e^x$ üstel fonksiyonudur.	>> <b>exp</b> (1) ans = 2.7183
<b>log(x)</b>	Doğal logaritma (ln) fonsiyonudur	>> <b>log</b> (2.7183) ans = 1
<b>log10(x)</b>	10 tabanında logaritma foksiyonudur.	>> <b>log10</b> (10) ans = 1
<b>sqrt(x)</b>	Karekök fonksiyonudur.	>> <b>sqrt</b> (4) ans = 2
<b>nthroot(x)</b>	n. dereceden kök fonksiyonudur.	>> <b>nthroot</b> (16,4) ans = 2
<b>realpow(x)</b>	Gerçek sayıların kuvvetini hesaplar.	>> <b>realpow</b> (2,2) ans = 4



# İŞLEÇLER ÜÇE AYRILIR: ARİTMETİK, İLİŞKİSEL VE MANTIKSAL

## İLİŞKİSEL İŞLEÇLER:

- < KÜÇÜKTÜR
- <= KÜÇÜK EŞİTTİR
- > BÜYÜKTÜR
- >= BÜYÜK EŞİTTİR
- == EŞİTTİR
- ~= EŞİT DEĞİLDİR

## MANTIKSAL İŞLEÇLER:

- ~ NOT (DEĞİL)
- & AND (VE)
- | OR (VEYA)

# MATLAB PROGRAMLARINA KLAVYEDEN VERİ AKTARIMI

"**input**" komutu yardımıyla bir **MATLAB** programı içerisinde değerlendirilmek üzere kullanıcıdan klavye yoluyla bir veri alınır ve bu veri bir değişkene atanır.

Command Window

```
>> yas=input('Lutfen Yasinizi Giriniz: ')
```

```
Lutfen Yasinizi Giriniz: 27
```

```
yas =
```

```
27
```

Veri Girişi Sembolü

Bardak en fazla %90 oranında dolsun

```
>> Bardak=input('Bardağın Ne Kadarı Dolsun? ')
```

```
Bardağın Ne Kadarı Dolsun? 90
```

```
Bardak =
```

```
90
```

# input KOMUTU İLE KLAVYEDEN METİNSEL VERİ TEMİNİ

```
>> isim=input('Lutfen Isminizi Giriniz = ', 's' )
```

Command Window

```
>> isim=input('Lutfen Isminizi Giriniz: ' )  
Lutfen Isminizi Giriniz: Bora  
Error using input  
Undefined function or variable 'Bora'.
```

```
Lutfen Isminizi Giriniz: 27
```

```
isim =
```

```
27
```

```
>> isim=input('Lutfen Isminizi Giriniz: ','s')  
Lutfen Isminizi Giriniz: Bora
```

```
isim =
```

```
Bora
```

**%c** : *degerin tek bir karakter* olduğunu gösterir.  
**%s** : *degerin bir karakter dizisi (string)* olduğunu gösterir.

**%d** : *degerin bir tamsayı* olduğunu gösterir.

**%f** : *degerin bir ondalıklı sayı* olduğunu gösterir.

**%g** : *degeri* mümkün olan en kompakt forma sokar.

Diğer yandan :

**\n** : İmleci bir alt satırın başına götürür.

**(n, newline)**

**\t** : İmleci bir TAB kadar sağa kaydırır.

# FPRINTF KOMUTU İLE EKRANA BİLGİ YAZDIRMA

**fprintf**('Ekrana Basılacak Açıklama %X \n', *değer* );

Command Window

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

## Komutu İle Ekranaya Sayısal Değer Yazdırma

```
disp('Üzgünüm! Sıfıra Bölüm Hatası Var.');
```

```
fprintf('Üzgünüm! Sıfıra Bölüm Hatası Var.\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.



# DISP KOMUTU İLE EKRANA SAYISAL DEĞER YAZDIRMA

```
Command Window
>> skaler=16;
>> satirVektoru=[12 -4 36 25 47];
>> matris=[1 2;3 4];
>> skaler

skaler =

    16

>> disp(skaler);
    16

>> disp(satirVektoru);
    12    -4    36    25    47

>> disp(matris);
     1     2
     3     4
```

## MATLAB/Uygulama-2

$$\mathbf{B} = \begin{bmatrix} 10 & 5 & 5 \\ 70 & 8 & 7 \\ 10 & 1 & 3 \end{bmatrix}$$

Aşağıdaki işlemleri command window'da yapınız.

- 1) B matrisini giriniz.
- 2) B matrisini mevcut çalışma klasörünüze **katsayilar** ismiyle kaydediniz.
- 3) Dosyanın kaydedilip kaydedilmediğini kontrol ediniz. (Open Files penceresinden)
- 4) MATLAB oturumundaki tüm değişkenleri siliniz (clear)
- 5) Command window'da yazılmış tüm ifadeleri temizleyiniz. (clc)
- 6)  $B*2$  işlemini yapınız.
- 7) B matrisini geri çağırınız.
- 8) B matrisinin üst ve alt üçgen matrislerini oluşturunuz.
- 9)  $C=[B \text{ zeros}(3,2)]$  işlemini yapınız.

# MATLAB/Uygulama-2:Çözüm

1 >> B=[10 5 5;70 8 7;10 1 3];

2 >> save katsayilar B

4 >> clear

5 >> clc

6 >> B\*2

??? Undefined function or variable 'B'. **Neden?**

7 >> load katsayilar

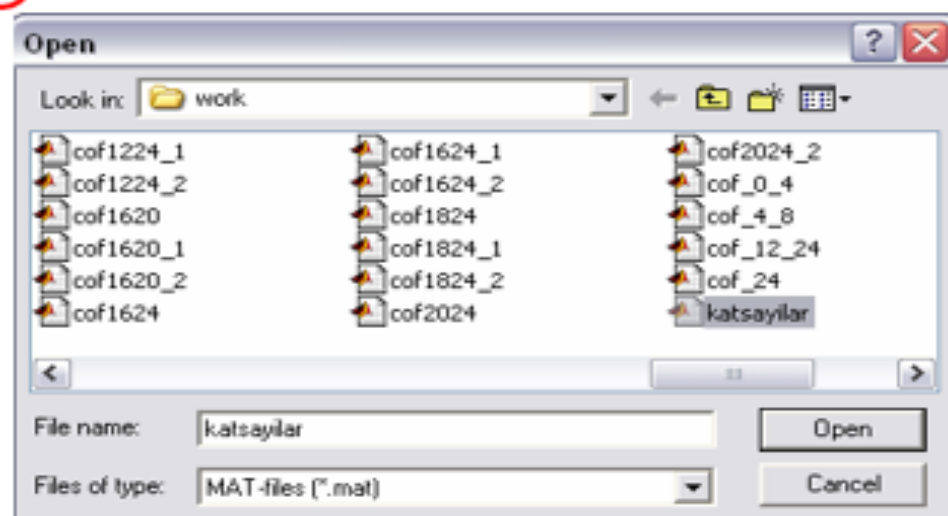
8 >> triu(B)

```
ans =  
    10     5     5  
     0     8     7  
     0     0     3
```

>>tril(B)

```
ans =  
    10     0     0  
    70     8     0  
    10     1     3
```

3



9

>> C=[B zeros(3,2)]

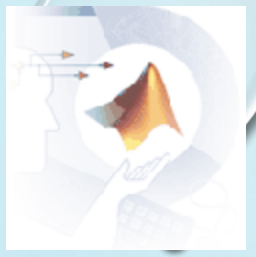
```
C =  
    10     5     5     0     0  
    70     8     7     0     0  
    10     1     3     0     0
```



# MATLAB PROGRAMLAMA

Bilgisayar programlamasında genel olarak belirli kalıp ve kurallara uyulur. Bir bilgisayar yazılımının oluşturulmasında genel olarak aşağıda sıralanan prosedüre uyulur:

- Problemin tanımı (Ne-Niçin)
- Çözüm yönteminin ve adımlarının belirlenmesi (algoritma: akış şemaları veya pseudo-kod)
- Kodlama (Programlama diline çevirme)
- Sınama (Programı çalıştırma)
- Belgeleme ve Güncelleştirme



# MATLAB DİZİLER

## DİZİLER

- DİZİ, EN GENEL MATEMATİKSEL TANIMI İLE NÜMERİK VE METİNSEL DEĞERLER TOPLULUĞUDUR. MATLAB'DE HERŞEY BİR DİZİ OLARAK İŞLEME KONUR VE EN TEMEL VERİ ELEMANIDIR.
  - REEL İLE KOMPLEKS SAYILARI İFADE EDEN ÇİFTKAT VEYA NÜMERİK DİZİLER (DOUBLE VEYA NUMERIC ARRAY)
  - METİN İFADE EDEN DİZİLER (CHAR ARRAY)
  - HÜCRE DİZİLER (CELL VEYA STRUCT ARRAY)

## MATLAB/DİZİLER (ARRAYS)

**Sayılar** : Sayı dizileri (numeric array)  
**Karakterler** : Karakter dizileri(character array)

```
c=2017 (numeric array)
d='İstanbul Üniversitesi' (character array)
f=[2017 2018] (numeric, matrix)
g=[d 'mühendislik
Fakultesi'] (character, matrix)
```

**Hücreler** : Hücre dizileri(cell array)  
**Yapılar** : Yapı dizileri (structure array)

**Not: Numeric ile character dizileri bir matriste bir arada bulunamaz! Yani, bir matris hem sayı hem de bir kelimeyi aynı anda içeremez!**

# MATLAB/Diziler (Arrays)

**`C=cell(n)`**  $n \times n$  hücreden oluşan boş bir hücreyi **C**'ye atar.

Örneğin  $n=2$  için

**>> C=cell(2)** C =

```
    []    []  
    []    []
```

**hücreleri oluşturulur.**

**Bir hücrenin içine istenilen sayıda yeni hücreler eklemek mümkündür;**

**Örneğin,**

**`C{1}{1}=[2 3]` ile **C** aşağıdaki biçimde değişir;**

**C =**

```
    {1x1 cell}    []  
    []           []
```

**Yapı dizileri (Structure arrays), veri tabanları için oldukça kullanışlı bir dizi türüdür.**

**A yapı dizisi çağrıldığında,**

```
A.name='galatasaray';  
A.sname='spor kulübü';  
A.city='Istanbul';  
A.email='galatasaray1905@gmail.com';  
A.year=1905;
```

```
>>A  
  
A =  
    name: 'galatasay'  
    sname: 'spor kulübü'  
    city: 'Istanbul'  
    email: 'galatasaray1905@gmail.com'  
    year: 1905
```

**ile A, bir structure array olur.**

**Hücre ve yapı dizileri, mat uzantılı dosyalar olarak, daha önce açıklanan save komutuyla kaydedilip, load komutuyla geri çağrılabilir.**





**num2str(a)**  
**string)**

Bir a sayısını bir karaktere atama (From **numeric** to (**2**)

**str2num(a)**

Karakter olan bir a sayısını sayı değerine atama

**mat2str(a)**

Bir a matrisini bir karakter dizisine atama

**int2str(a)**

Bir a tam sayısını bir karaktere atama

**char(a)** Bir a hücrelerini bir karakter dizisine atama

**cellstr(a)**

Bir a karakterini bir hücre dizisine atama

**num2cell(a)**

Bir a sayısını bir hücre dizisine atama

# MATLAB/İF,END YAPISI

**if** (eğer) yapısı bir koşulun gerçekleşmesi durumunda bir işlemi yaptırmak için sıklıkla kullanılır.

Bu ifade,

**if** koşul

işlem

**end** biçimindedir.

**Örnek:** Girilen bir sayının negatif olması durumunda, sayıyı doğal logaritmasıyla değiştiren bir kod düşünelim:

```
a=input(' bir sayi  
giriniz= ');  
if a<0  
    a=log(a);  
else  
    a=a;  
end  
a
```

“Diğer durumda”  
anlamındadır:  
Burada,  $a \geq 0$   
koşulunu temsil  
eder.

## Else yapısı kullanılmıyaydı

```
a=input(' bir sayi giriniz= ');  
if a<0  
    a=log(a);  
end  
if a>0  
    a=a;  
end  
a
```

## MATLAB/SWİTCH,CASE YAPISI

**switch** (değiştir) if yapısına benzer. Burada daha çok sözel olarak belirtilen durumlara göre yönlendirme işlemi yapılır. Bu yapının kullanımı **case** ile aşağıdaki gibidir;

```
switch durum
  case durum
    işlem1
  case durum2
    işlem2
  otherwise
    işlem3
end
```

otherwise  
işlem3

→ Kullanımı kişiye bağlıdır.

**Örnek:** gun degiskeninin, is gunu olup olmadigina karar vermek için aşağıdaki kodlar düşünülür;

```
clear,clc
gun=input('hangi gun=', 's');
switch lower(gun)
  case{'pazartesi','sali','carsamba','persembe','cuma'}
    disp('işgünü')
  case {'cumartesi','pazar'}
    disp('TATİL!')
end
```

# MATLAB/FOR,END DÖNGÜSÜ

**for,end** döngüsü bir işlemin birden daha fazla sayıda yaptırılmasında kullanılır.

(Örneğin, kök bulma problemlerinde kullanılan iterasyon çözümleri).  
Kullanımı,

```
for i=1:n      (i→Tam sayı (integer))  
    işlem  
end
```

biçimindedir.

**Örnek:** 1'den N'ye kadar olan sayıların toplamını yapan bir program düşünelim.

```
clear,clc  
N=input('bir sayı giriniz=');  
say=0; %sayac  
for i=1:N  
    say=say+i; %birikimli (kümülatif toplam)  
end  
disp(say)
```

## MATLAB/while,end döngüsü

`while,end` döngüsü, belirli bir durumun gerçekleşmesi durumunda bir işlemin birden daha fazla sayıda yaptırılmasında kullanılır.

```
done=0;  
while done==0  
    işlem  
end
```

1. Buradaki, `while,end` döngüsü, `done` değişkeni ancak ve ancak 0 olduğu zaman çalışacaktır.
2. Bir önceki satırda, `done` değişkeni 0 olarak atanmış olduğu için `while,end` döngüsü çalışır.  
(*while, end döngüsünü çalıştıran farklı algoritmalara burada değinilmeyecektir.*)

**Örnek:** 1'den N'ye kadar olan sayıların toplamını `while,end` döngüsü ile yapan bir program düşünelim.

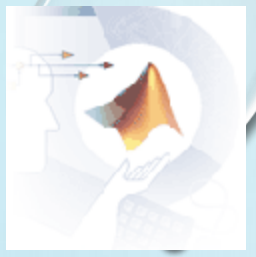
```
clear,clc  
N=input('bir sayı giriniz=');  
say=0;i=0;done=0;  
while done==0
```

```
    i=i+1; %bir önceki örnekte for,end döngüsündeki "i" ye karşılık gelir.
```

```
    if i==N  
        done=1;  
    end
```

```
    say=say+i;  
end  
say
```

`i`, son sayıya (N'ye) ulaştığında, `done` değişkenine 0'dan farklı bir sayı atanır. Böylece, `while`'ın olduğu satıra gelindiğinde, `done` "0" olmadığı için `while, end` döngüsü çalışmaz (döngü sonlanır). Program, bu döngünün `end` satırının hemen altındaki satırdan işleme devam eder (burada, `say` değişkeni `command window`'da yazdırılır.).



# KAYNAKÇA

Dođan, U., (2009), Temel Bilgisayar Bilimleri Ders Notları, YTÜ, Lisans Ders Notları, İstanbul.

Demirel, H., (2005), Dengeleme Hesabı, YTÜ, Lisans Ders Notları, İstanbul.

Uzunođlu M., vd. (2002), Matlab, Türkmen Kitabevi, İstanbul.

<http://www.mathworks.com/matlabcentral/>

<http://www.mathworks.com/matlabcentral/fileexchange/>

Matlab İle Programlama (Dr. Deniz DAL)