

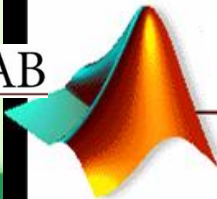
MATLAB

2.DERS



MATLAB'DE DİZİLER

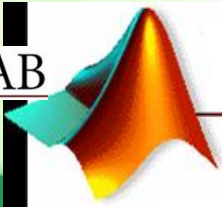
- Dizi (array), en genel tanımı ile **nümerik** veya **metinsel değerler topluluğudur**. (**veri yapısı – data structure**) MATLAB'de herşey bir dizi olarak işleme konur ve dizi en temel veri elemanıdır.
 - Reel ile kompleks sayıları ifade eden çift kat veya **nümerik diziler** (double veya numeric array)
 - Metin ifade eden diziler, **karakter dizileri** (char array)





MATLAB'DE DİZİLER (devam)

- Bir nümerik dizi, **skaler**, **vektör** veya **matris** olabilir ve tüm nümerik diziler **double array** formatındadır.
- **1x1** dizisi, bir **skaler** (scalar) gösterir. (a=3, b=-6.5)
- **mx1** veya **1xn** dizisi, bir **vektör** (vector) gösterir.
- **mxn** veya **nxm** dizisi, bir **matris** (matrix) gösterir.
- Bu çerçevede **1x1** dizisi sabit matris veya **tek elemanlı matris**, **nx1** dizisi **sütun matrisi** ve **1xn** dizisi ise **satır matrisi** olarak da düşünülebilir.
- Bir dizinin eleman sayısı, satır ile sütun sayısının çarpımıdır.



MATLAB'DE DİZİLER (devam)

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

3x2 matrix → 6 eleman

$$b = [1 \quad 2 \quad 3 \quad 4]$$

1x4 array → 4 eleman, **satır vektörü**

$$c = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$$

3x1 array → 3 eleman, **sütun vektörü**

$$A(2,2)=4$$

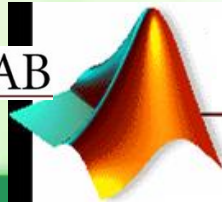
Satır #

Sütun #

$$b(3)=3$$

$$c(1)=1$$

MATLAB



VEKTÖRLER

1xn veya **mx1** şeklinde tek boyutlu dizi olan vektörleri oluşturmanın iki temel yolu vardır:

i) Direkt olarak ([...] ile)

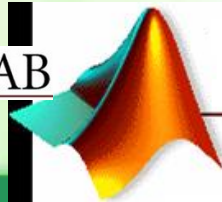
- **Satır vektörleri:** `>> f = [f1 f2 f3 ... fn]` veya
`>> f = [f1,f2,f3, ...,fn]`
- **Sütun vektörleri:** `>> f = [f1; f2; f3; ...; fn]`

ii) Eşit aralıklı elemanlar kullanarak (: ile)

- **f = İlkDeğer : DeğişimMiktarı : SonDeğer**

Değişim miktarı belirtilmezse İlkDeğer'den sonra 1'er artım olacağını ifade eder.

ÖRNEK : a=1:10 veya b=1:5:25



MATRİSLER

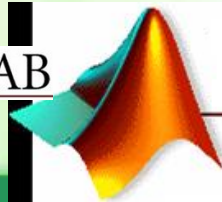
$$F = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \dots & \cdot & \cdot & \dots \\ \dots & \cdot & \cdot & \dots \\ F_{m1} & F_{m2} & \cdot & F_{mn} \end{bmatrix}_{m \times n}$$

Matrisleri oluşturma en temel yolu [...] kullanmaktır.
mxn boyutundaki bir matrisin genel formatı:

- $F = [F_{11} \ F_{12} \ \dots \ F_{1n} ; F_{21} \ F_{22} \ \dots \ F_{2n} ; \dots ; F_{m1} \ F_{m2} \ \dots \ F_{mn}]$

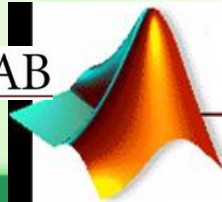
veya

- $F = [F_{11}, F_{12}, \dots, F_{1n} ; F_{21}, F_{22}, \dots, F_{2n}; \dots ; F_{m1}, F_{m2}, \dots, F_{mn}]$



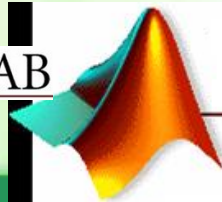
DİZİ EDITÖRÜ (ARRAY EDITOR)

Çalışma alanında (**workspace**) herhangi bir değişkenin üzerine çift tıklarsanız **Microsoft Excel** tablosuna (spreadsheet) benzer bir pencere açılır sizin için. Buna **dizi editörü** ya da **array editor** denir. Değişkeninizin elemanlarını bu pencere yardımıyla da modifiye edebilirsiniz.



FONKSİYONLAR

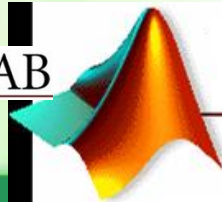
- Fonksiyonlar da bilgisayar programlarıdır. Bilgisayarlar yardımıyla çözülmeye çalışılan problemler fonksiyonlar sayesinde yönetilmesi daha kolay, küçük parçacıklara bölünürler. Her fonksiyon:
 - Kendine özgü bir isme sahiptir.
 - Kendine, üzerinde işlem yapacağı bir **argüman** (parametre) ya da **argümanlar** (parametreler) alır.
 - GENELLİKLE geriye bir **değer döndürür**. (skaler, vektör ya da matris.)
 - **Örnek : rand(n,m)**



MATLAB'DE DİZİLER (devam)

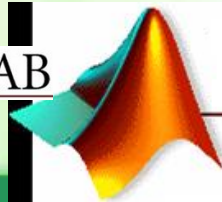
Hem vektörler hem de matrisler yardımcı (utility) fonksiyonlar (**zeros**, **ones** ve **rand**) kullanılarak da oluşturulabilir:

- **zeros**(1,n) veya **zeros**(n,1)
- **zeros**(n) veya **zeros**(n,n)
- **zeros**(n,m)
- **ones**(1,n) veya **ones**(n,1)
- **ones**(n) veya **ones**(n,n)
- **ones**(n,m)
- **rand**(n,m)
- **rand**(n) veya **rand**(n,n)
- **round**(**rand**(n,m))
- **fix**(**rand**(n,m)) (**Nasıl bir çıktı ????**)



MATLAB'DE RASTGELE SAYI ÜRETİMİ

- **rand(n,m)** fonksiyonu MATLAB'de **nxm** boyutunda ve elemanlari **rastgele sayilar** olan bir matris olusturur. Olusturulan bu rastgele sayilar **0 ile 1** arasindadir ve duzgün (uniformly) dagilimlidir.
- Rastgele sayilardan olusan bu matrisin tum elemanlarini **k** gibi bir sayi ile carparak sayilari **0 ile k** araligina cekebiliriz.
- Ondalik sayilardan olusmus bir matrisin elemanlarini yuvarlayip tam sayi yapmak icin **round** fonksiyonunu kullanabiliriz.
- **a=rand(1,10) ??**
- **b=round(40*rand(5,3)) ?? c= 40*round(rand(5,3)) ??**
- **d=round(10+40*rand(5,3)) ?? e=10+40*round(rand(5,3))??**
- **f=round(50+250*rand(3,4)) ?? g=round(1000*rand(1)) ??**

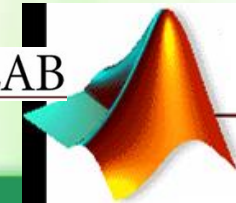


DİZİLERE UYGULANABİLEN BAZI FONKSİYONLAR

- **length(a):** **a** vektörünün eleman sayısı
- **sum(a):** **a** vektörünün elemanlarının toplamı
- **max(a):** **a** vektörünün maksimum elemanı
- **min(a):** **a** vektörünün minimum elemanı
- **size(b):** **b** matrisinin boyutu (satır ve sütun)
- **size(b,1):** **b** matrisinin satır sayısı
- **size(b,2):** **b** matrisinin sütun sayısı
- **sum(b):** **b** matrisinin sütun başına eleman toplamı (!!! Satır Vektörü!!!)
- **sum(sum(b)):** ?????????
 - **a(n):** **a** vektörünün n. eleman değeri
 - **b(n,:):** **b** matrisinin sadece n. satırı
 - **b(:,n):** **b** matrisinin sadece n. sütunu
 - **b':** **b** matrisinin transpozesi yani devriği
- **det(b):** **b** matrisinin determinantı !!!! (Kare matrisler için)!!!!
- **inv(b) :** **b** matrisinin tersi !!!!(Kare matrisler için)!!!!
- **diag(b):** **b** matrisinin ana diagonal (çapraz) elemanları
- Elemanter işlemler eleman eleman işlem demektir ve **.** işleci kullanılır:
Çarpma: **.*** , Bölme: **./** , Üs alma: **.^**

Örnek : Bir matrisin tüm elemanlarının karesini alma ($a^2=a*a$) ????

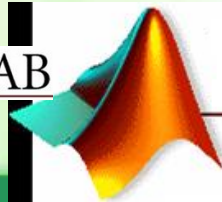
MATLAB



UYGULAMA

- **Soru:** MATLAB'ın **size** fonksiyonunu kullanarak bir **a** vektörünün boyutunu nasıl bulursunuz?

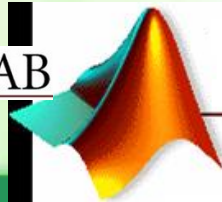
- **Cevap:** $\max(\text{size}(\mathbf{a})) = \text{length}(\mathbf{a})$



MATLAB'DA PROGRAMLAMA

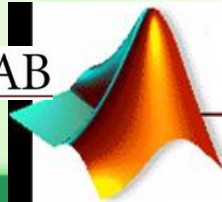
Bilgisayar programlamasında genel olarak belirli kalıp ve kurallara uyulur. Bir bilgisayar yazılımının oluşturulmasında genel olarak aşağıda sıralanan prosedüre uyulur:

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



MATLAB'DA PROGRAMLAMA

- MATLAB'da programlama en genel olarak iki yolla yapılır:
 - Komut satırında (**in-line**) programlama
 - m-dosyalarıyla (**m-files**) programlama
- m-dosyalarının da iki türü vardır:
 - Düzyazı (**script**) m-dosyaları
 - Fonksiyon (**function**) m-dosyaları
- m dosyaları oluşturabilmek için bir metin editörüne ihtiyaç vardır.

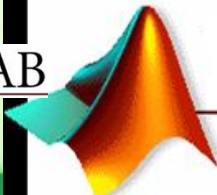


MATLAB' da Programlama

M-dosyası (M-File)

Bir senaryo dosyası (script file) özel bir görevi yerine getirmek için gerekli **MATLAB** komutlarının saklandığı bir metin programıdır. Başka bir ifadeyle; bir hesaplamayı gerçekleştirmek için yazılacak bir çok komut dizisi, komut penceresinden tek tek girmek yerine bir dosyada saklanır daha sonra bu dosya çalıştırılarak bu komutlar icra edilir. Bu dosyaların **MATLAB**'in çalıştığı dizinde (current directory) "**dosya_adi.m**" uzantısıyla saklanmaları gerekir. **DOSYA ADLARININ İÇİNDE TÜRKÇE KARAKTERLER VE BOŞLUKLAR KULLANILAMAZ.**

Senaryo dosyalarının (**M-dosyalarının**) oluşturulması ve yazılması için **MATLAB** bir metin hazırlayıcısı (text editor) sunmaktadır. Bu senaryo dosyaları Windows'da **Notepad** gibi herhangi bir metin hazırlayıcısında da yazılabilirler. **MATLAB** metin hazırlayıcısı ya "**current directory**" penceresinde boş bir alana sağ tıklayıp "**New, M-File**" ile ya da kısaca "**File**" menüsünden "**New, M-File**" ibaresini seçerek etkin hale getirilebilir.



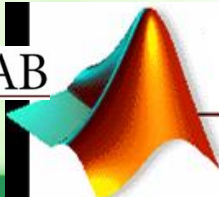
M-dosyalarının Gerekliliği:

- ✓ Değişken sayısının fazla olması durumunda
- ✓ Akış diyagramlarının uygulanmasında
- ✓ Programdaki değişikliklerin kolayca yapılmasında

Örnek:

Dışarıdan klavye yoluyla girilen dairenin yarıçapına göre **alanini** ve **çevresini** hesaplayan bir MATLAB programını “**alanVeCevreHesapla.m**” adında bir **M-dosyasi** içinde yazınız ve komut penceresinden “**alanVeCevreHesapla**” komutunu yazarak çalıştırınız.

```
clc;  
clear;  
r = input('Dairenin yarıçapını giriniz = ');  
alan=pi*r^2;  
cevre=2*pi*r;  
alan,cevre
```



MATEMATİKSEL VE MANTIKSAL OPERATÖRLER

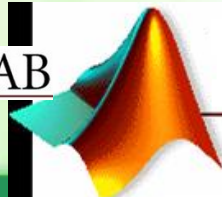
Program dallandıran bir çok yapıda, işlemler, sonucun "**doğru**" (**TRUE**) (**1**) veya "**yanlış**" (**FALSE**) (**0**) olması durumları ile kontrol edilir. MATLAB'de '**doğru**' veya '**yanlış**' ifadeleri ile sonuçlar üreten **iki çeşit** operatör vardır. Bunlar **matematiksel** ve **mantıksal** operatörlerdir. (Bir önceki derste **aritmetik** operatorleri gormustuk. Dolayısıyla MATLAB'de su ana kadar 3 tip operator gormus olduk.)

Matematiksel Karşılaştırma Operatörleri

Bu operatörler iki değişkenin değer bakımından karşılaştırmasını yaparlar ve üretilen sonucun **doğru** (**1**) veya **yanlış** (**0**) durumuna göre sonuçlar üretirler.

Genel kullanımları **a1 işlem a2** şeklindedir. Burada a1 ve a2, **aritmetik değerler**, **değişkenler** veya **karakter dizileri** olabilir, "**işlem**" ise, sözkonusu matematiksel kıyaslama operatörlerinden biri olabilir. Eğer a1 ve a2 arasındaki ilişki operatörün belirttiği şekilde ise **sonuç 1** değerini alır. Eğer operatörün belirttiği durumdan farklı bir durum söz konusu ise **sonuç 0** değerini alır.

==	Eşittir	5==6	→	0
~=	Eşit değildir	5~=6	→	1
>	Büyüktür	5>6	→	0
>=	Büyük veya eşittir	5>=6	→	0
<	Küçüktür	5<6	→	1
<=	Küçük veya eşittir	5<=6	→	1



Eşitlik durumlarında verilen işaret iki adet eşittir "==" işaretinden oluşur. Oysa değişken atamalarında kullandığımız eşittir "=" bir tanedir. Bu ikisi birbirlerinden farklı operatörlerdir. "==" operatörü, kıyaslama durumlarında kullanılır ve mantıksal bir sonuç üretir. "Eşit ise", "eşit midir?" şeklindeki durumlarda kullanılır. "=" işareti ise, bir değişkene bir değer atamada (atama operatörü) kullanılır, örneğin **MATLAB** komut penceresinde; 3=5 yazdığımızda; program hata üretir. Oysa 3==5 yazdığımızda bu "3, 5'e eşit midir?" anlamına gelir, kıyaslama yapar ve **MATLAB** bu durum için "0" cevabını üretir. Yeni başlayanlar için bir karşılaştırma durumunda tek eşittir "=" işareti kullanmak, sık yapılan bir hatadır.

```
>> 3==5
ans =
    0
```

```
>> 3 = 5
```

```
??? 3=5
```

```
!
```

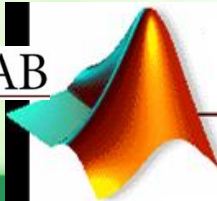
Error: The expression to the left of the equals sign is not a valid target for an assignment.

```
>> x=0;
>> y=sin(pi);
```

şeklinde yapılan iki değişken atamasını takiben yapılan x==y karşılaştırmasının sonucu olarak **1** cevabı beklenirken, **0** cevabı alınır. **MATLAB**, bu iki değeri farklı olarak algılamıştır. Çünkü **MATLAB**'da sin(pi) değeri 0'a eşit değildir, sin(pi), yuvarlamadaki farklılıktan ötürü 1.2246e-016 değerine sahiptir ve **0**'a eşit değildir. Yani teorik olarak birbirine eşit iki değer, aslında farklı sayılara tekabül etmektedir.

```
>>x=0;
>>y=sin(pi);
>>x==y
ans = 0
```

MATLAB



Sık yapılabilecek diğer bir hata da **karşılaştırma** operatörlerinin **aritmetik** operatörlerden **daha sonra** değerlendirildikleri durumunu ihmal etmektir. Yani parantezlerden yararlanılmadığı durumlarda bile aritmetik işlemler, **öncelikle** yapılır.

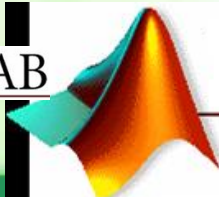
$$2+8 > 8+3$$

$$(2+8) > (8+3)$$

Bu iki durumda da **MATLAB**'in üreteceği cevap **0**'dır.

Mantıksal Operatörler

Bu operatörler, bir veya iki mantıksal anlamı olan ve mantıksal bir sonuç üreten operatörlerdir. Üç tane çiftli mantık operatörü vardır. Bunlar; "**AND**", "**OR**" ve "**XOR**" dur. Diğer bir mantık operatörü tekli yapıya sahip olan "**NOT**" operatörüdür. Çiftli yapıdaki operatörlerin genel kullanımı **a işlem b** şeklinde iken tekli bir operatör olan "**NOT**" genel kullanımı **işlem a** şeklindedir. Burada **a** ve **b** **değişkenler** iken **işlem**, aşağıdaki mantık operatörlerinden biridir, **a** ile **b**'nin arasındaki ilişki, operatörün belirttiği şekilde ise sonuç **1**, eğer değilse sonuç **0** olur.



Operatörler	Komut karşılığı	İşlevleri
a&b	AND	VE
a b	OR	VEYA
xor(a,b)	XOR	Özel Veya
~a	NOT	Değil

Örnek:

```
>> k=4; m=5;
```

```
>> (k>6) and (m<8)      →    HATALI YAZIM
```

```
??? (k>6) and (m<8)
```

```
Error: Unexpected MATLAB expression
```

```
>> (k>6) & (m<8)
```

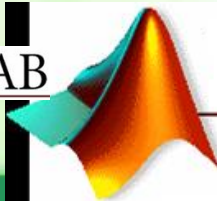
```
ans =
```

```
0
```

0<=x<9 ifadesinin
MATLAB'deki karşılığı:

(0<=x) & (x<9)

MATLAB



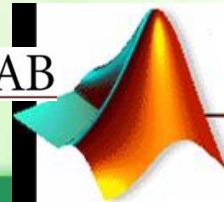
MANTIKSAL OPERATORLERIN DOGRULUK TABLOLARI

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	xor(A,B)
0	0	0
0	1	1
1	0	1
1	1	0

A	~A
0	1
1	0





Kontrol

Yapıları

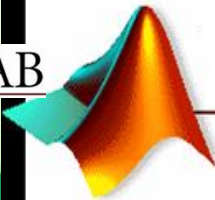
IF

Switch

For Loop

While Loops

MATLAB



if Şartlı deyimi (Conditional Statement)

Bir mantıksal ifadeyi kontrol ederek bunun sonucuna göre mümkün seçeneklerden birini icra edebilen bir komuttur.

if Deyiminin Üç Farklı Kullanım Sekli vardır.

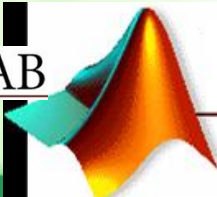
```
if Şart
    1. işlem
    2. işlem
    3. işlem
    :
end
```

```
if Şart
    1. işlem;
else
    2. işlem
end
```

```
if Şart
...
elseif ...
...
else
...
end
```

Şart dediğimiz şey bir karşılaştırma ifadesi (**a>b gibi**) ya da bir mantıksal ifadedir (**a&b gibi**).

MATLAB



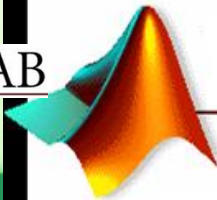
Örnek: Girilen (okutulan) x ve y değerlerine göre aşağıdaki sonuc değerini bulan bir MATLAB programını bir M-dosyası içine yazınız ve komut penceresinden dosya adı ile çalıştırınız.

x > y ise $\text{sonuc} = \sqrt{x-y}$
x = y ise $\text{sonuc} = (x+y)^7$
x < y ise $\text{sonuc} = x*y$

ÇÖZÜM:

```
clc;  
clear;  
x=input('x değeri=');  
y=input('y değeri=');  
if x>y  
    sonuc=sqrt(x-y)  
elseif x==y  
    sonuc=(x+y)^7  
else  
    sonuc=x*y  
end
```

MATLAB



Uygulama:

Dışarıdan girilen **x** değerlerine göre aşağıdaki fonksiyonun değerini hesaplayan bir **MATLAB** programını bir M-dosyası içine yazınız ve komut penceresinden dosya adı ile çalıştırınız.

$$1 \leq x < 10$$

$$F(x) = x + x^2 + x^3 + x^4 + x^5$$

$$10 \leq x$$

$$F(x) = x + x^2 + \frac{\ln(x) + 1}{\log_{10}(x)} + \tan(x) + \sqrt{x} + \frac{3 \cdot x + 5}{x^4}$$

