

MATLAB Kursu

HAZIRLAYAN: BURAK ÖZPOYRAZ - burakozpoyraz@gmail.com

İçindekiler

DERS - 1.....	1
1) MATLAB VE YAZILIM TEMELLERİ.....	1
1.1) MATLAB Nedir?.....	1
1.2) Programlama Nedir?.....	2
2) MATLAB ARAYÜZÜ.....	2
3) MATLAB UYGULAMALARI.....	3
4) DEĞİŞKEN TANIMLAMA VE ARİTMETİK İŞLEMLER.....	3
4.1) Değişken Tanımlama.....	3
4.2) Aritmetik İşlemler.....	3
4.3) Hazır Fonksiyonlar (Built-In Functions).....	3
4.4) Vektörler.....	5
5) PROBLEM / DERSİMİZ OLASILIK VE İSTATİSTİK.....	7

DERS - 1

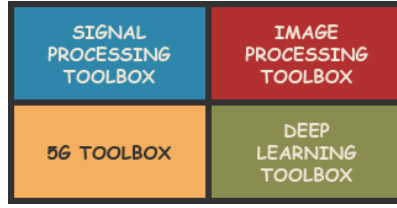
1) MATLAB VE YAZILIM TEMELLERİ

1.1) MATLAB Nedir?



- **Matrix Laboratory**
- Hesaplamalar için kullanılan yüksek performanslı bir yazılım dilidir.
- Teknik hesaplamaları ve veri görselleştirmeyi birleştirerek çok fonksiyonlu programlar tasarlanabilir.
- Yazımı oldukça basit bir dili vardır.
- Endüstride yapılan AR-GE çalışmalarında, ürün geliştirme süreçlerinde ve teknik analiz aşamalarında sıkça kullanılan bir yazılım ortamıdır.

- Akademide yapılan arařtırmaların simülasyon ařamalarında, yüksek performansta matematiksel hesaplama ve analiz gerektiren derslerde sıklıkla kullanılmaktadır.
- Kullanıcılar açısından çok faydalı olan **toolbox**'lar kullanıma hazır olarak bulunmaktadır. Örnek olarak:



- Birçok ders artık MATLAB üzerinden tasarlanacak projeler barındırıyor. Dersleri çalışırken interaktif olarak MATLAB'i kullanın. Bu sizin MATLAB'e olan alışkanlığınızı arttıracaktır.

1.2) Programlama Nedir?

- Yazılım dili öğrenmek → 1 Gün
- Program yazabilme yeteneđi → Proje yaptıkça oluşur.
- Algoritma tasarlayabilme beceresi kazanmak programlama yeteneđinin oluşması için çok önemlidir.



- Kendinizi zorlayacak bir problem ile karşılařtıđınızda onu programlayarak çözmeye çalışın, bu sizin algoritma becerinize katkı sağlayacak.
- Bir dilin tüm fonksiyonları tamamen bilinemez. Bir işlemi gerçekleřtirecek fonksiyonu internetten mutlaka arařtırın.
- Hata mesajı okumayı ve internette bu hatayı aratmayı öğrenmeniz gerekiyor.
- Stack Overflow
- MathWorks: MATLAB Cody, Self-Paced Online Courses

ÖNEMLİ: Sizin yaptığınız hatayı daha önce yapmış birisi veya karşılařtıđınız probleme çözüm aramış birisi **MUTLAKA VARDIR.**

2) MATLAB ARAYÜZÜ

- Command Window
- Workspace
- Editor
- Live Editor
- MATLAB Online
- MATLAB Grader

3) MATLAB UYGULAMALARI

- Matematiksel hesaplamalar
- Algoritma geliştirme
- Veri toplama (Data Acquisition)
- Modelleme
- Simülasyon
- Veri analizi (Data Analysis)
- Veri görselleştirme (Visualization)
- Arayüz tasarımı

4) DEĞİŞKEN TANIMLAMA VE ARİTMETİK İŞLEMLER

4.1) Değişken Tanımlama

```
a = 4 % Value is shown in the command window.  
b = 7; % b is defined here but is redefined below, so it warns us with orange  
% underline.  
b = 8;  
b = ;% SYNTAX ERROR  
name = "Burak";  
13;
```

4.2) Aritmetik İşlemler

```
c = 7 + 21;  
d = 12 - 8;  
e1 = 5 * 9;  
g = 64 / 4;  
h = 17 / 5;  
i = 2^6;  
k = 3 * (23 + 14.7 - 8) / 7.2;  
l = 3 == 5;
```

4.3) Hazır Fonksiyonlar (Built-In Functions)

```
j = 5i;  
pi_val = pi;  
sqrt_val = sqrt(25);  
exp_val = exp(3);  
log_val = log10(100);  
ln_val = log(exp(1));
```

EXAMPLE

"e" sayısı hazır olarak bulunmadığına göre, kolay kullanım için "e" sayısı nasıl tanımlanabilir?

```
e = exp(1);
```

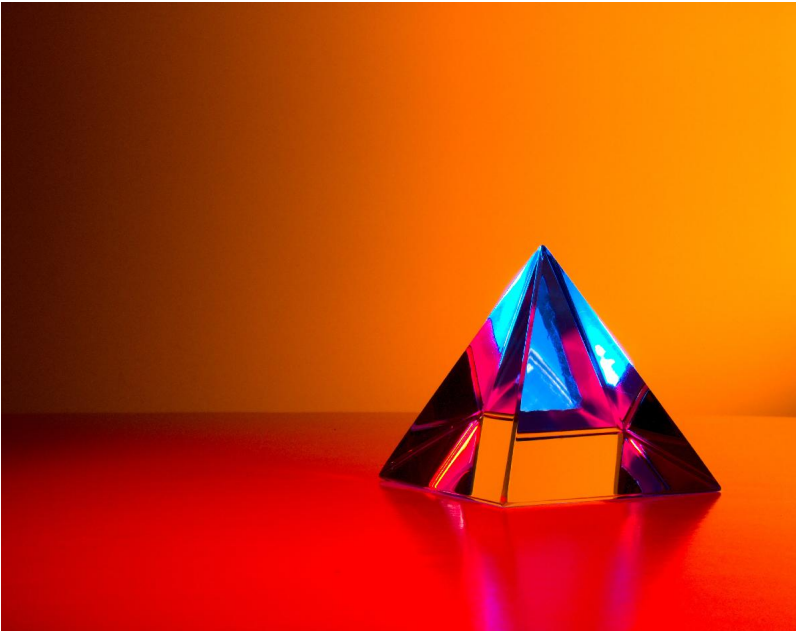
```
log_val2 = log5(25);
```

EXAMPLE

Hazır fonksiyonlar arasında "log5(x)" bulunmadığına göre, $\log_5 25$ değeri nasıl bulunabilir?

```
log_5_25 = log(25) / log(5);
```

```
sin_val = sin(pi / 2);  
cos_val = cos(pi);  
tan_val = tan(pi / 4);  
arc_sin_val = asin(1) / pi;  
arc_cos_val = acos(-1) / pi;  
arc_tan_val = atan(1) / pi;  
abs_val = abs(-17);  
angle_val = angle(1 + 1i) / pi;  
conj_val = conj(1 + 3i);  
z = 3 + 5i;  
real_z = real(z);  
imag_z = imag(z);  
fac_val = factorial(4);  
comb_val = nchoosek(4, 2);  
modulo = mod(5, 3);  
ceil_val = ceil(8.2);  
floor_val = floor(12.6);  
round_val = round(5.8);
```



4.4) Vektörler

```
prime = [2 3 5 7 11 13];  
length_row_vector = length(prime);  
size_row_vector = size(prime);  
  
vector = [1, 3, 5, 7, 9];
```

MATLAB'de vektörlerin indisleri 1'den başlar. Birçok yazılım dilinde bu hususta farklılıklar yaşandığından dikkat edilmelidir. Bir vektörün herhangi bir indisindeki elemanına aşağıdaki gibi erişilebilir.

```
element_at_1 = prime(1);  
element_at_4 = prime(4);  
prime(6) = 17;  
element_at_7 = prime(7); % What do you expect as the result?  
prime(9) = 19; % What do you expect as the result?
```

Eşit aralıklara sahip elemanlardan oluşan bir vektör aşağıdaki gibi oluşturulabilir.

```
vector2 = 1 : 2 : 12;  
vector3 = 2 : 4 : 12;  
vector4 = 11 : -2 : 1;  
vector5 = 1 : 0.2 : 2;  
vector6 = 1 : -2 : 3; % What do you expect as the result?
```

Eşit aralıklara sahip belli bir eleman sayısından oluşan bir vektör aşağıdaki gibi oluşturulabilir.

```
vector7 = linspace(-7, 7); % 100 evenly spaced points  
vector8 = linspace(-7, 7, 15); % 15 evenly spaced points
```

Şimdiye kadar gördüğümüz vektörlerin hepsi satır vektördür. Sütun vektörler ise aşağıda verildiği gibi oluşturulabilir.

```
col_vector1 = [2; 3; 5];
```

Sütun vektörü, aynı zamanda satır vektörünün **transpozu** alınarak da oluşturulabilir. Gerçek sayılardan (real numbers) oluşan bir satır vektörünün transpozu aşağıdaki gibi alınabilir.

```
row_vector = [7 12 90];  
col_vector2 = transpose(row_vector);  
col_vector3 = row_vector';  
col_vector4 = row_vector.');
```

ÖNEMLİ: Kesme işareti normalde **konjuge transpoze** almak için kullanılır. Ancak gerçek sayıların konjugeleri kendilerine eşit olduğu için gerçek sayılarda normal transpoz almaya karşı düşer. Ancak kesme ile transpoz almayı alışkanlık haline getirirseniz sonrasında imajiner sayıların olduğu programlarda sıkıntı yaşayabilirsiniz. O yüzden **transpose** fonksiyonunu veya **.'** operatörünü kullanmak en garanti yoldur.

Aynı sayıda elemana sahip olan vektörler aritmetik işlemlere tabi tutulabilirler.

```
a_vector = [2 4 5 -3 9 -12];  
b_vector = [3 7 8 16 -21 4];  
addition_vector = a_vector + b_vector;  
subtraction_vector = a_vector - b_vector;  
matrix_product = a_vector * b_vector; % What do you expect as the result?  
a_vectorx2 = 2 * a_vector;  
sum_val = sum(a_vector);  
prod_val = prod(a_vector);  
diff_vector = diff(a_vector);  
index = find(a_vector == 9);
```

Eğer bir hazır fonksiyonun ismini biliyor ama ne yaptığını tam bilmiyor veya hatırlamıyorsanız, iki şekilde öğrenebilirsiniz.

1. Fonksiyonun dokümantasyonuna MathWorks sayfasından bakabilirsiniz.
2. **help** komutunu kullanabilirsiniz.

```
help sum
```

Element bazında (element-wise) çarpma ve bölme işlemleri aşağıda verildiği gibi yapılabilir.

```
c_vector = [1 2 3 4 5 6];  
d_vector = -5 : 2 : 5;  
multiplication = c_vector .* d_vector;  
division = c_vector ./ d_vector;  
power = c_vector.^2;
```



EXAMPLE

İlk 6 asal sayının karelerinin toplamını bulunuz.

```
prime6 = [2 3 5 7 11 13];  
result = sum(prime6.^2);
```

MATLAB'in hazır fonksiyonları normal değişkenlere uygulandığı gibi aynen vektörlere de uygulanabilir. Bu durumda fonksiyon vektördeki her elemana ayrı ayrı uygulanarak sonuçta yine bir vektör elde edilir.

```
angle_vector = 0 : 10 : 90;  
rad_vector = angle_vector * pi / 180;  
sin_vector = sin(rad_vector);
```

Mühendislikte vektörlerin istatistiksel verileri oldukça önemlidir. Bu veriler, MATLAB'de bulunan hazır fonksiyonlar sayesinde aşağıda verildiği gibi hesaplanabilir.

```
e_vector = [1 3 5 7 9];  
mean_val = mean(e_vector);  
var_val = var(e_vector);  
min_val = min(e_vector);  
max_val = max(e_vector);
```

5) PROBLEM / DERSİMİZ OLASILIK VE İSTATİSTİK



5 öğrencinin bulunduğu bir "Olasılık ve İstatistik" dersinde 2 vize ve 1 final yapılmaktadır. Buna ek olarak da dönem boyunca 2 ödev verilmektedir. Öğrencilerin ödev, vize ve final notlarını içeren tablo aşağıda verilmiştir:

	EMİR	EMRE	SELİN	BATUHAN	CANSU
ÖDEV-1	80	37	90	20	95
ÖDEV-2	75	45	88	52	100
VİZE-1	90	70	80	61	100
VİZE-2	70	3	64	47	76
FİNAL	85	70	95	80	90

Öğrencinin dersten aldığı ortalama notu belirlemek için ödevlerin her birisinin %5'i, vizelerin her birisinin %25'i ve finalin de %40'ı alınmaktadır.

1) Her öğrencinin sonuçlarını farklı bir vektör haline getiriniz.

```
emir = [80 75 90 70 85];  
emre = [37 45 70 3 70];  
selin = [90 88 80 64 95];  
batuhan = [20 52 61 47 80];  
cansu = [95 100 100 76 90];
```

2) Her öğrencinin ders ortalama notunu hesaplayarak **avg_vec** isminde yeni bir vektör elde ediniz.

```
weight_vec = [0.05 0.05 0.25 0.25 0.4];  
avg_vec(1) = sum(emir .* weight_vec);  
avg_vec(2) = sum(emre .* weight_vec);  
avg_vec(3) = sum(selin .* weight_vec);  
avg_vec(4) = sum(batuhan .* weight_vec);  
avg_vec(5) = sum(cansu .* weight_vec);
```

3) Sınavların zorluklarını görebilmek adına vizelerin ve finalin ayrı ayrı ortalamalarını bularak **mean_vec** isminde yeni bir vektör elde ediniz.


```

midterm1 = [90 70 80 61 100];
midterm2 = [70 3 64 47 76];
final = [85 70 95 80 90];

mean_vec(1) = mean(midterm1);
mean_vec(2) = mean(midterm2);
mean_vec(3) = mean(final);

```

4) Emre 2. vizeden aldığı sonucu çok düşük bularak itiraz etmiştir. Bunun sonucunda Emre'nin kağıdını yeniden inceleyen hoca bir soruda hata yaptığını farketmiş ve notu 32 olarak değiştirmiştir. Bu durumda Emre'nin notlarını tutan, her öğrencinin ders ortalama notunu tutan ve vizeler ile finalin ortalama değerlerini tutan vektörleri güncelleyiniz.

```

emre(4) = 32;
avg_vec(2) = sum(emre .* weight_vec);
midterm2(2) = 32;
mean_vec(2) = mean(midterm2);

```

5) Öğrencilerin ikinci vizeden düşük aldığını gören hoca, 2. vize özelinde çan uygulamaya karar vermiştir. Bunun için en yüksek not alan öğrencinin notu 100'e yükseltirken diğer öğrencilerin notları da aynı büyüklükte değiştirilmiştir. Bu durumda öğrencilerin notlarını tutan, her öğrencinin ders ortalama notunu tutan ve vizeler ile finalin ortalama değerlerini tutan vektörleri güncelleyiniz.

```

max_midterm2 = max(midterm2);
diff_point = 100 - max_midterm2;

emir(4) = emir(4) + diff_point;
emre(4) = emre(4) + diff_point;
selin(4) = selin(4) + diff_point;
batuhan(4) = batuhan(4) + diff_point;
cansu(4) = cansu(4) + diff_point;

avg_vec(1) = sum(emir .* weight_vec);
avg_vec(2) = sum(emre .* weight_vec);
avg_vec(3) = sum(selin .* weight_vec);
avg_vec(4) = sum(batuhan .* weight_vec);
avg_vec(5) = sum(cansu .* weight_vec);

midterm2 = midterm2 + diff_point;
mean_vec(2) = mean(midterm2);

```

6) Bu derste alınan tüm notların standart sapmasını aşağıda verilen formülü uygulayarak hesaplayınız ve varyans değerinin kareköküne eşit olduğunu teyit ediniz. Sonrasında ortalama ve medyan değerleri arasındaki farkı hesaplayınız ve standart sapma ile beraber sınıfın performansını yorumlayınız.

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - m)^2}$$

```
all_notes = [emir emre selin batuhan cansu];
N = length(all_notes);
m = mean(all_notes);
sigma = sqrt(sum((all_notes - m).^2) / (N - 1));
equality = sigma == sqrt(var(all_notes));

all_notes = sort(all_notes);
even_length_median = mean(all_notes(N / 2 : (N / 2) + 1));
odd_length_median = all_notes(ceil(N / 2));
median = (mod(N, 2) == 0) * even_length_median +...
        (mod(N, 2) == 1) * odd_length_median;
```