



MATLAB ile ANALİZ (MIA)

DERS-7



FONKSİYONLAR (nargin - nargout)

Bir fonksiyonda giriş parametrelerinin kontrolü

Nargin = number of argument input

```
function q = targ( a,b,c )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
if (nargin==1)
    q=a.^2;
elseif (nargin==2)
    q=a.^2+b.^2;
elseif (nargin==3)
    q=a.^2+b.^2+c.^2;
end
end
```



FONKSİYONLAR (nargin - nargout)

Kullanıcıları giriş parametrelerini tam girmeye zorlamak...

```
function q = targ2( a,b,c )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
if (nargin<3)
    disp('En az üç parametre girilmelidir.');
```

$$q = a^2 + b^2 + c^2$$

```
elseif (nargin==3)
    q=a.^2+b.^2+c.^2;
end
end
```



FONKSİYONLAR (nargin - nargout)

Bir fonksiyonda çıkış parametrelerinin kontrolü, **nargout**

```
function [elemanSay, art_o, geo_o]=ortalamalar(x)
if nargin==1
    elemanSay=length(x);
elseif nargin==2
    elemanSay=length(x);
    art_o=mean(x);
elseif nargin==3
    elemanSay=length(x);
    art_o=mean(x);
    geo_o=geomean(x);

end
```



FONKSİYONLAR (handle)

Bir dosyanın içine birden çok fonksiyon yazarak fonksiyon tutamaçlarını (*handle*) döndürme işlemi.

```
function f = fonksiyonlar()  
    f.f1 = @fonksiyon1;  
    f.f2 = @fonksiyon2;  
end
```

```
function y = fonksiyon1(x)  
    y = x^2;  
end
```

```
function z = fonksiyon2()  
    z = 5;  
end
```

Fonksiyonları aşağıdaki şekilde çağırabiliriz.

```
>> f.f1(2)
```

```
ans =
```

```
    4
```

```
>> f.f2()
```

```
ans =
```

```
    5
```



find() fonksiyonu

find(x) = Bir matris ya da dizide x ifadesinin 'True' yani 1 olduğu elemanların indeksini verir.

```
>> x = [-8 0 -1 3 4.5];  
find(x)
```

```
ans =
```

```
1 3 4 5
```

Ayrıca find komutu içinde verilen koşulu sağlayan elemanların indekslerini de verir.

```
>> find(x>1)
```

```
ans =
```

```
4 5
```



find() fonksiyonu

```
>> X = [1 0 2; 0 1 1; 0 0 4]
```

```
X =
```

```
1 0 2  
0 1 1  
0 0 4
```

```
>> k=find(X)
```

```
k =
```

```
1  
5  
7  
8  
9
```

```
>> k2 = find(~X)
```

```
k2 =
```

```
2  
3  
4  
6
```

```
>> X(k2)
```

```
ans =
```

```
0  
0  
0  
0
```



find() fonksiyonu

```
>> r=rastgele12(-20,20,5,5)
```

```
r =
```

```
13 -16 -14 -14 6  
16 -9 19 -3 -19  
-15 2 18 17 14  
17 18 -1 12 17  
5 19 12 18 7
```

```
>> r(1:3:25)=0
```

```
r =
```

```
0 -16 -14 0 6  
16 0 19 -3 0  
-15 2 0 17 14  
0 18 -1 0 17  
5 0 12 18 0
```

```
>> find(r,2)
```

```
ans =
```

```
2  
3
```

```
>> find(r,2,'last')
```

```
ans =
```

```
23  
24
```




find() fonksiyonu

r =

```
0 -16 -14 0 6
16 0 19 -3 0
-15 2 0 17 14
0 18 -1 0 17
5 0 12 18 0
```

>> k=find(r<0 & r>-15)

k =

```
11
14
17
```

>> [satir,sutun]=find(r<0 & r>-15)

satir =

```
1
4
2
```

sutun =

```
3
3
4
```



find() fonksiyonu

$$x^3 - 4.2x^2 + 3.3x - 4$$

Fonksiyonunun $x=-5:0.0001:5$ aralığındaki köklerini find komutu ile bulunuz.

```
x=-5:0.0001:5;
```

```
>> y=x.^3-4.2*x.^2+3.3.*x-4;
```

```
kokIndis=find(y>=-0.001 & y<=0.001)
```

```
kokIndis =  
85913      85914
```

```
>> y(kokIndis)
```

```
ans =
```

```
1.0e-03 *
```

```
-0.5616    0.6209
```

```
>> x(kokIndis)
```

```
ans =
```

```
3.5912    3.5913
```



reshape() fonksiyonu

Matlab'da $A(m \times n)$ boyutunda bir matris $m \times n = p \times q$ olmak şartıyla $B(p \times q)$ boyutunda bir matrise dönüştürülebilir.

Örnek: A matrisi (2×3) boyutlarındadır. A matrisini (3×2) boyutuna getiriniz. $A = [10 \ 8 \ 6 ; 1 \ 3 \ 5];$

```
>> B=reshape(A,3,2)
```

B =

```
10  3
 1  6
 8  5
```



reshape() fonksiyonu

```
>> r=rastgele12(-20,20,1,20)
```

```
r =
```

```
Columns 1 through 12
```

```
10 10 -4 6 -13 8 -19 -9 -18 -16 13 8 -7 18 -19 -2 -5 11 12 -13
```

```
>> r1=reshape(r,5,4)
```

```
r1 =
```

```
10 8 13 -2  
10 -19 8 -5  
-4 -9 -7 11  
6 -18 18 12  
-13 -16 -19 -13
```



reshape() fonksiyonu

>> r3=reshape(r,[],2) İki sütunlu olsun.
Satır sayısı farketmiyor. Yeter ki tam olarak
2 sütun yapılabilsin.

r3 =

```
10 13
10  8
-4 -7
 6 18
-13 -19
 8 -2
-19 -5
-9 11
-18 12
-16 -13
```

>> r3=reshape(r,[],3)

Error using reshape Product of known dimensions, 3, not
divisible into total number of elements, 20.



eval() - feval() fonksiyonları

Girdi olarak fonksiyon kabul eden fonksiyonlardır. Bazıları:
Fplot, feval, fzero, fminsearch vs.

Fplot fonksiyonuna grafikler konusunda bakılacak.

eval ve feval fonksiyonları:

Eval(): Bir karakter dizisini işleme koyar. Sanki komut satırından komut satırından bir komut girmiş gibi işlem yapar.

```
>> x=eval('cosd(45)')
```

```
x =
```

```
0.7071
```

```
>> eval('x=1');
```

```
.
```



eval() - feval() fonksiyonları

```
>>eval('1+1')  
>>eval('x=1+1')
```

Feval() ise bir karakter dizisi yerine bir fonksiyona işlem yapar

```
>> x=feval('cosd',135)
```

```
x =  
-0.7071
```



eval() - feval() fonksiyonları

```
>>fzero('cosd', [0 180])
```

```
ans =
```

```
90
```

Örnek: $x^3 - 4.2x^2 + 3.3x - 4$ Fonksiyonunun kökünü bulunuz.

```
>> fzero('x.^3-4.2*x.^2+3.3.*x -4',[-5 5])
```

```
ans =
```

```
3.5912
```

```
>> fzero('x.^3-4.2*x.^2+3.3.*x -4',1)
```

```
ans =
```

```
3.5912
```




Karakter dizisi fonksiyonları (string fonksiyonlar)

MATLAB tüm değişkenleri diziler olarak kabul eder ve string değişkenler karakter dizileri olarak kabul edilir.

Değişkeni kontrol etmek için **whos** komutunu kullanalım

```
>> gir='program'
```

```
gir =  
program
```

```
>> whos gir
```

Name	Size	Bytes	Class	Attributes
gir	1x7	14	char	



Karakter dizisi fonksiyonları (string fonksiyonlar)

- Oluşturma ve Bitiştirme Fonksiyonları
- Bul – Değiştir Fonksiyonları
- Birleştirme ve Ayırma Fonksiyonları
- Düzenleme Fonksiyonları
- Karşılaştırma Fonksiyonları



Karakter dizisi fonksiyonları (string fonksiyonlar)

```
>> gir='program'  
gir =  
program  
  
>> x='1886'  
x =  
1886  
  
>> y=[gir,' kodu: ',x]  
y =  
program kodu: 1886  
  
>> y(4:9)  
ans =  
gram k
```

```
>> z='Ali'nin'  
z =  
Ali'nin  
  
>> d=double(z)  
d =  
    65  108  105  39  110  105  
110
```

```
>> y=[gir;' kodu: ';x]  
Error using vertcat  
  
>> strvcat(gir,' kodu: ',x)  
ans =  
program  
kodu:  
1886  
  
>> strcat(gir,' kodu: ',x)  
ans =  
program kodu:1886
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

Oluşturma ve Bitiştirme Fonksiyonları

strcat(s1,s2,..sn) stringleri birleştiren fonksiyon.

```
>> s1='AKU';  
>> s2='teknoloji';  
>> s3='fakültesi';  
>> strcat(s1,s2,s3)  
ans =  
AKUteknolojifakültesi
```

```
>> str1 = {'John ','Mary '};  
>> str2 = {'Smith','Jones'};  
>> str = strcat(str1,str2)
```

str =

```
'John Smith' 'Mary Jones'
```

```
>> str = strcat('Dr. ',str)
```

str =

```
'Dr.John Smith' 'Dr.Mary Jones'
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

```
>> A={'resim1.jpg','resim2.jpg','cameraman.jpg','lina.jpg','chessboard.jpg'}
```

```
A =
```

```
'resim1.jpg' 'resim2.jpg' 'cameraman.jpg' 'lina.jpg' 'chessboard.jpg'
```

```
>> B=strjoin(A,',')
```

```
B =
```

```
resim1.jpg,resim2.jpg,cameraman.jpg,lina.jpg,chessboard.jpg
```

```
>> B=strjoin(A,'-')
```

```
B =
```

```
resim1.jpg-resim2.jpg-cameraman.jpg-lina.jpg-chessboard.jpg
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

sprintf() karakter dizisi içindeki bilgileri düzenler. fprintf() 'den farkı bir değişkene atanabilmesidir.

Örnek:

```
A1 = 2;
```

```
A2 = 3;
```

```
str = sprintf('Dizinin boyutları: %dx%d',A1,A2)
```

```
str2=strcat(str,' işlemleri yapınız!')
```

```
str2 =
```

```
Dizinin boyutları: 2x3. işlemleri yapınız!
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

Bul – Değiştir Fonksiyonları

strfind (str,aranan)

```
A=['resim1.jpg, ','resim2.png, ','cameraman.tif, ','lina.jpg, ','chessboard.png']
```

```
A =  
resim1.jpg, resim2.png, cameraman.tif, lina.jpg, chessboard.png
```

```
>> strfind(A,'png')
```

```
ans =
```

```
20 61
```

```
>> findstr(A,'png')
```

```
ans =
```

```
20 61
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

strrep(karakter dizisi, aranan, yerine gelecek olan)

```
>> a='AKÜ Otomotiv Mühendisliği'
```

```
a =  
AKÜ Otomotiv Mühendisliği  
  
>> b=strrep(a,'Otomotiv','Mekatronik')
```

```
b =  
AKÜ Mekatronik Mühendisliği
```

```
>A={'resim1.jpg';'resim2.jpg';'cameraman.jpg';'lina.jpg';'chessboard.jpg'}
```

```
A =
```

```
'resim1.jpg'  
'resim2.jpg'  
'cameraman.jpg'  
'lina.jpg'  
'chessboard.jpg'
```

```
>> strrep(A,'jpg','tiff')
```

```
ans =
```

```
'resim1.tiff'  
'resim2.tiff'  
'cameraman.tiff'  
'lina.tiff'  
'chessboard.tiff'
```




Karakter dizisi fonksiyonları (string fonksiyonlar)

```
>> gezi_tarihleri={'Başlangıç:___'; 'Bitiş:___'}
```

```
gezi_tarihleri =
```

```
 'Başlangıç:___'
```

```
 'Bitiş:___'
```

```
>> tarihler={'11/11/2016';'13/11/2016'}
```

```
tarihler =
```

```
 '11/11/2016'
```

```
 '13/11/2016'
```

```
>> tam_tarih=strrep(gezi_tarihleri,'___',tarihler)
```

```
tam_tarih =
```

```
 'Başlangıç:11/11/2016'
```

```
 'Bitiş:13/11/2016'
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

strcmp(karakter dizisi1, karakter dizisi2)

Büyük-küçük harf duyarlıdır.

strcmpi(karakter dizisi1, karakter dizisi2)

Büyük-küçük harf duyarlı değildir.

```
>> a='AKÜ Mekanik'
```

```
a =  
AKÜ Mekanik
```

```
>> b='AKÜ mekanik'
```

```
b =  
AKÜ mekanik
```

```
>> strcmp(a,b)
```

```
ans =  
0
```

```
>> b='AKÜ Mekanik'
```

```
b =  
AKÜ Mekanik
```

```
>> strcmp(a,b)
```

```
ans =
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

```
>> a={'AKÜ', 'Mekatronik', 'Mühendisliği'}
```

```
a =
```

```
'AKÜ' 'Mekatronik' 'Mühendisliği'
```

```
>> b={'AKÜ', 'mekatronik', 'Mühendisliği'}
```

```
b =
```

```
'AKÜ' 'mekatronik' 'Mühendisliği'
```

```
>> strcmp(a,b)
```

```
ans =
```

```
1 0 1
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

strsplit(karakter dizisi)
strsplit(karakter dizisi, ayırıcı)

Karakter dizilerini belirli bir karaktere göre ayırmaya yarar.

```
>> a='AKÜ Mekantronik Mühendisliği'
```

```
a =
```

```
AKÜ Mekantronik Mühendisliği
```

```
>> b=strsplit(a)
```

```
b =
```

```
'AKÜ' 'Mekantronik' 'Mühendisliği'
```

```
>> a='AKÜ_Mekantronik_Mühendisliği'
```

```
a =
```

```
AKÜ_Mekantronik_Mühendisliği
```

```
>> b=strsplit(a,'_')
```

```
b =
```

```
'AKÜ' 'Mekantronik' 'Mühendisliği'
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

Karakter – Sayı Dönüşümleri

str2num() fonksiyonu

str2double() fonksiyonu

```
>> b=str2num('144')
```

```
b =  
144
```

```
>> b+1
```

```
ans =  
145
```

```
>> a=str2num('abc')
```

```
a =  
[]
```

```
>> c=str2num('144+10')
```

```
c =  
154
```

```
>> d=str2num('144 +10')
```

```
d =
```

```
>> e='1.66e-2'
```

```
e =  
1.66e-2
```

```
>> str2num(e)
```

```
ans =  
0.0166
```

```
>> str2double(e)
```

```
ans =  
0.0166
```



Karakter dizisi fonksiyonları (string fonksiyonlar)

Say - Karakter Dönüşümleri

int2str() fonksiyonu

```
>> b=144  
b =  
144
```

num2str() fonksiyonu

```
>> a=int2str(b)  
a =  
144
```

```
>> a=num2str(b)  
a =  
144
```

```
>> b=144.33  
b =  
144.3300
```

```
>> a=int2str(b)  
a =  
144
```

```
>> a=num2str(b)  
a =  
144.33
```



DOSYA İŞLEMLERİ

Şu ana kadar bir programda hesaplanan veya elde edilen veriler RAM'de saklanacak şekilde tanımlanmıştı. Yani, bilgisayar kapatıldığında veya MATLAB programı sona erdirildiğinde veriler de kaybolmaktaydı.

Oysa pek çok uygulamada, elde edilen verilerin daha sonra kullanılmak üzere saklanması gerekir. Bunun için bu verilerin bir dosyaya yazılması gerekmektedir.



DOSYA İŞLEMLERİ

1. İşleyeceğin dosyayı aç. (fopen)
2. Açtığın dosyayı oku ya da açtığın dosyaya yaz. (fscanf, fprintf)
3. Açtığın dosyayı kapat. (fclose)



DOSYA İŞLEMLERİ

MATLAB'deki fopen komutu, bilgi kaydetmek veya bilgi okumak için, varolan bir veri dosyasını yazmaya/okumaya açar veya varolmayan bir dosyayı sıfırdan oluşturur.

Yazım formatı:

dosya_degiskeni = fopen('dosya adi','mod')

Burada:

dosya_degiskeni-> Dosya kontrolünde kullanılan MATLAB değişkeni

dosya adi-> Verilerin kaydedileceği/okunacağı dosyanın adı

mod-> Dosya işleme modu : r,w,..., vs.



DOSYA İŞLEMLERİ

MOD	AÇIKLAMA
'r'	Dosyayı sadece okumaya açar. (yazmaya izin vermez.)
'r+'	Dosyayı okumaya ve yazmaya açar.
'w'	Varolan bir veri dosyasının içindekini siler, dosya yoksa oluşturur ve dosyayı yazmaya açar.
'w+'	Varolan bir veri dosyasını yazmak için açar, dosya yoksa oluşturur ve dosyayı okumaya ve yazmaya açar.
'a'	Varolan bir veri dosyasını yazmak için açar, dosya yoksa oluşturur ve girilecek bilgileri dosya sonuna ekler. (append)
'a+'	Varolan bir veri dosyasını okumak ve yazmak için açar, dosya yoksa oluşturur ve girilecek bilgileri dosya sonuna ekler. (append)



DOSYA İŞLEMLERİ

Dosya Kapama: `fclose`

Yazma ve/veya okumaya açılan dosyanın MATLAB'den ilişkisini kesmeye (dosyayı kapamaya) yarayan komuttur.

Kullanımı:

`fclose(dosya_degiskeni)`



DOSYA İŞLEMLERİ

```
% dosya yazma örneği  
x=100*rand(30,1);  
dosyaID=fopen('ornek11.txt','w')  
;  
fprintf(dosyaID,'%2.4f\r\n',x);  
fclose(dosyaID);  
%%  
%dosya okuma örneği  
dosyaID=fopen('ornek1.txt','r');  
A=fscanf(dosyaID,'%f')  
fclose(dosyaID);  
%%
```



DOSYA İŞLEMLERİ

```
%dosya yazma (iki sutun)
x=1:5;
y=[x;rand(1,5)];
dosyaID=fopen('ornek25.txt','w');
fprintf(dosyaID,'%d %f\r\n',y);
fclose(dosyaID);
%%
%dosya okuma (iki sutun)
dosyaID=fopen('ornek22.txt','r');
B=fscanf(dosyaID,'%d %f',[2 Inf]);
fclose(dosyaID);
%%
```



Karakter Dizisi Fonksiyonları

Örnek:

Bir grup sensörden veri okunmaktadır. Bu sensörler: potansiyometre, ldr (ışık sensörü) ve lm35'dir.

Gelen veriler 0-5V aralığında 3 (+-)1 V değerlerinde 200 elemanlı bir dizi oluşturulacaktır. Bu dizi:

POT_3.7196 LDR_2.8871 LM35_1.9977 POT_2.8853 ... şeklinde olacaktır.

«_» karakterinden sonraki veriler sensörlerden okunan voltaj değerleri olup random olarak belirlenecektir. «_» karakteri öncesindeki değerler ise voltaj değerinin okunduğu sensörü belirlemektedir.



Karakter Dizisi Fonksiyonları

```
clear;
format bank
r1=3*ones(1,200);
r2=-1+2*rand(1,200);
r=r1+r2;
% olcumler=zeros(1,200);
for i=1:200
olcumler{i}=strcat('POT_',num2str(r(i)));
end
%%
for k=2:3:200;
olcumler{k}=strrep(olcumler{k},'POT','LDR');
end

%%
for k=3:3:200;
olcumler{k}=strrep(olcumler{k},'POT','SCK');
end
```



Karakter Dizisi Fonksiyonları

Örnek:

Bir önceki örnekte oluşturulan:

POT_3.7196 LDR_2.8871 LM35_1.9977 POT_2.8853 ...

Diziden potD, ldrD, sckD isminde 3 ayrı dizi oluşturulacaktır.

Başında POT olan elemanların sağ yani sayı kısmı potD dizinine,

Başında LDR olan elemanların sağ yani sayı kısmı ldrD dizinine,

Başında LM35 olan elemanların sağ yani sayı kısmı sckD dizinine,

aktarılacaktır.



Karakter Dizisi Fonksiyonları

```
j=1;
for i=1:200
    d=strsplit(olcumler{i}, '_');
    if strcmp(d(1), 'POT')
        potD(j)=str2double(d(2));
        j=j+1;
    end
end
```

```
j=1;
for i=1:200
    d=strsplit(olcumler{i}, '_');
    if strcmp(d(1), 'LDR')
        ldrD(j)=str2double(d(2));
        j=j+1;
    end
end
```

```
j=1;
for i=1:200
    d=strsplit(olcumler{i}, '_') ;
    if strcmp(d(1), 'SCK')
        sckD(j)=str2double(d(2));
        j=j+1;
    end
end
```