



# GÖRÜNTÜ İŞLEME MATLAB

DERS-4

## SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

### For ve İf örnekleri

$X^3 - 4.2x^2 + 3.3x - 4$  fonksiyonunun köklerini bulunuz.

```
tic
for x=-5:0.0001:5
    t=x.^3-4.2*x.^2+3.3.*x -4;
    %   if(t==0)
    if (t>=-0.001 && t<=0.001)
        fprintf('Kök=%2.5f',t);
        fprintf(' değer=%2.5f\n',x);
    %       disp(t);
    end
end
plot(x,t,'LineWidth',3)
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
toc
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

$$f(x) = 2x - \cos x$$

Fonksiyonun köklerini döngüler kullanarak bulunuz.

```
tic
clc
clear
for x=-5:0.0001:5
    t=2*x-cos(x);

    if (t>=-0.0001 && t<=0.0001)
        fprintf('Kök=%2.5f',x);
        fprintf(' deđer=%2.5f\n',t);
    end
end
```

```
toc
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

1'den 999'a kadar olan sayılardan asal sayı olanların toplamını bulunuz.

### 1. yöntem:

```
total = 0;
for k = 1:999
if(isprime(k))
total = total + k;
end
end
disp(total)
```

### 2. yöntem

```
total = 0;
for k = primes(999)
total = total + k;
end
disp(total)
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

1'den 200'e kadar olan sayılardan kendisi ve 2 fazlası asal olan sayıları bulunuz.

```
for x = 1:2:200
if(isprime(x) & isprime(x+2))
fprintf('%f ve %f ikisi de asal sayıdır\n',x,x+2)
end
end
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

İç içe geçmiş döngüler (Nested Loops)

```
for x = 1:3
    for y = 1:2
        fprintf('x= %.0f and y= %.0f\n',x,y)
    end
end
```

```
x= 1 and y= 1
x= 1 and y= 2
x= 2 and y= 1
x= 2 and y= 2
x= 3 and y= 1
x= 3 and y= 2
```

## SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

Bir matrisi satır vektöre dönüştüren döngüyü oluşturunuz.

```
[m n] = size(A);  
k=1;  
for i=1:m  
    for j=1:n  
        v(k) = A(i,j);  
        k = k+1;  
    end  
end
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

### CELL ARRAY

Hem nümerik hem de nümerik olmayan değerler tek bir dizi ya da matriste ifade edilebilir mi?

Bazı durumlarda nümerik ve string ifadeler aynı dizi ya da matriste bulunabilmelidir.

```
w=[1 4 'aku' 5.3; 'afyon' -2.6 teknoloji' 1e2 ]
```

```
w={1 4 'aku', 5.3; 'afyon' -2.6 'teknoloji' ' 1e2 }
```

```
>>w(1,:)
```

```
>>isnumeric(w(1,1))
```

```
>>cell2mat(w(1,1))
```

```
>>[a b c ]=w{1,1:3}
```



# M-Fonksiyon Yapısı

## M-Fonksiyonlar kullanılırken dikkat edilecek hususlar:

- 1- Kullanıcılar kendi fonksiyonlarını yazmak için m-fonksiyonlarını kullanabilirler.
- 2- **Function** alt programı ve ana program şeklinde iki program yazılarak bu iki program ayrı ayrı kaydedilir.
- 3- Alt programdaki *fonksiyon\_adi*, m-dosyasına verilen isimle aynı olmalıdır.
- 4- Ana programdan alt program, function adı kullanılarak çağrılır.
- 5- Alt programdan da ana programa geçiş yapılabilir fakat genelde tercih edileni tersidir.
- 6- Parametre aktarımı olması durumunda alt ve ana programda eşit sayıda parametre ve giriş değişkeni olmalıdır.

**function** cikis\_ifadesi1, 2,..., n =**fonksiyon\_adi** (giris\_ifadesi1, 2, ...n)

**Örnek:** İki nokta arasındaki uzaklığı bulan programı m-fonksiyon (alt program) kullanarak yazınız.

x1=1.noktanın x koordinati; x2=2.noktanın x koordinati

y1=1.noktanın y koordinati; y2=2.noktanın y koordinati

**FUNCTION ALT PROGRAMI (uzak.m):**

**function uzaklik =uzak(x1,y1,x2,y2)**  
**uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);**

Bu function alt programı uzak.m olarak kaydedilir.

**ANA PROGRAM:**

**ax=3; ay=4; bx=1; by=2;**

**uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor**

**fprintf('iki nokta arasindaki uzaklık=%f',uzaklik);**

## Adım adım gerçekleştirilen işlemler:

- Ana program herhangi bir isimle kaydedilir ve koşturulur.
- Program, **function** adına (**uzak**) geldiği zaman alt program çağrılır ve ax, ay, bx, by parametreleri sırasıyla x1, y1, x2, y2 giriş değişkenlerine aktarılır.
- **Function** alt programında hesaplama gerçekleştirilir.
- **Function**'daki çıkış değişkeni olan **uzaklik** hem alt programda hem de ana programda hesaplanan sonuç değerinin aktarıldığı değişken olarak kullanılır.
- Alt programdan ana programa parametre aktarımı zorunlu değildir. İstenirse değişkenlerin değerleri alt programda da girilebilir ve sonuç alt programda yazdırılabilir.

**Uygulama:** Yukarıdaki örneği ana programdan alt programa parametre aktarımı yapmadan yeniden yazınız.

(Değişkenlerin girilmesi, sonucu hesaplama ve yazdırma işlemi alt programda yapılacaktır)

### FUNCTION ALT PROGRAMI:

```
function uzaklik = uzak(x1,y1,x2,y2)  
x1=3; y1=4; x2=1; y2=2;  
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);  
fprintf(`iki nokta arasindaki uzaklık=%f', uzaklik);
```

### ANA PROGRAM:

```
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
```

# FONKSİYONLAR

Rastgele sayı üreten bir fonksiyon yazınız.

```
function r = RastgeleSayi( altSinir, ustSinir, satir,  
sutun )  
%UNTITLED Summary of this function goes here  
% Detailed explanation goes here  
r = round(altSinir + (ustSinir-  
altSinir).*rand(satir,sutun)); % random neagtif, pozitif  
sayýlar  
  
end
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

Bir dizinin maksimum ve minimumunu bulan bir fonksiyon yazınız.

```
function [ maks, minimum ] = MaxMinBul( vektor )
```

```
maks=max(vektor);  
minimum=min(vektor);  
end
```

```
>> j=RastgeleSayi(-40,50,1,20);  
>> [maksimum, minimum]=MaxMinBul(j)
```

```
maksimum =
```

```
45
```

```
minimum =
```

```
-29
```

## SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

Bir vektör ya da matrisi [0-1] arasına normalize eden bir fonksiyon yazınız.

```
function y = NormalizeEt01(x)
%b alt sýnýr, a ust sýnýr
[satir,sutun]=size(x);
i=1; j=1;
for i=1:satir
    for j=1:sutun
        y(i,j)=[(x(i,j)-min(x(:)))]/[max(x(:))-min(x(:))];
    end
end
```