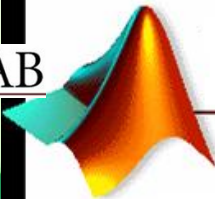


MATLAB

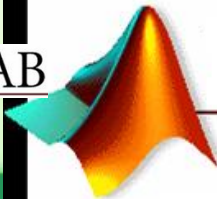
3.DERS

MATLAB



disp VEYA **fprintf** KOMUTLARIYLA EKCRANA MESAJ YAZDIRMA

Daha önceki derslerimizde **input** komutu ile klavye üzerinden MATLAB programlama ortamına veri girmeyi öğrenmiştik. Bu dersimizde ise **disp** veya **fprintf** komutları yardımıyla ekrana nasıl mesaj yazdırıldığını öğreneceğiz.



disp KOMUTUYLA EKRAMA MESAJ YAZDIRMA

disp ile Metinsel Çıkış

```
>>disp('Bu bir metin ciktisidir.')
```

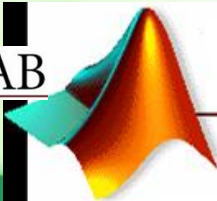
disp ile Nümerik Değişken Çıkışı

```
>>disp(numerik degisken)
```

disp ile Metinle Beraber Nümerik Değişken Çıkışı

```
>>disp(['Bu bir metin ciktisidir.' num2str(numerik degisken)])
```

ÖNEMLİ NOT : Bu slaytta “numerik degisken” dediğimiz şey bir **skaler**, bir **vektör** ya da bir **matris** olabilir.



fprintf KOMUTUYLA EKRA NA MESAJ YAZDIRMA

GENEL KULLANIM

fprintf('ekrana basılacak açıklama %X \n', deger)

Burada %X Kısmında Kullanabileceğimiz Seçenekler :

%c : degerin tek bir karakter olduğunu gösterir.

%s : degerin bir string olduğunu gösterir.

%d : degerin bir tam sayı olduğunu gösterir.

%f : degerin ondalıklı bir sayı olduğunu gösterir.

%e : degeri 10'un kuvveti üstel olarak gösterir.

%g : degeri yazılabilecek en kısa formda gösterir.

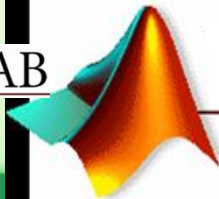
(Gereksiz sıfırlar atılır mesela.)

Diğer yandan :

\n : degeri ekrana yazdıktan sonra imleci bir satır atlatır.

\t : degeri ekrana yazdıktan sonra imleci bir TAB kadar atlatır.

\' : ekrana tek tırnak işareti basar.



ÖRNEK

Girilen iki sayının oranını bulan ve payda sıfır girildiğinde ekrana “uzgunum, sifira bolum hatasi var.” mesajı yazdıran program.

```
clc;  
clear;
```

```
pay = input('payi giriniz: ');  
payda = input('paydayi giriniz: ');
```

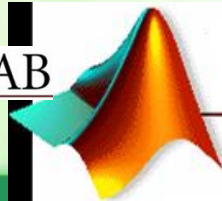
```
if payda==0  
    disp('uzgunum, sifira bolum hatasi var.');
```

else

```
    oran=pay/payda;  
    fprintf('oran = %f \n',oran);
```

end

$$\text{oran} = \frac{\text{pay}}{\text{payda}}$$

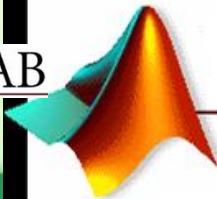


DÖNGÜLER (LOOPS)

Bir çok uygulamada bazı işlemlerin tekrar tekrar gerçekleştirilmesi gerekir. Bilgisayar programlama dillerinde, bu tür işlemleri çok sayıda tekrar etme imkanı sağlayan yapılara **ÇEVİRİM**, **DÖNGÜ** veya **LOOP** denir.

► Çevrim, bir tekrarlı işlem yapısıdır.

- ✓ **Çevrimdeki işlem sayısını önceden belirleyerek ve bu sayıya ulaşıp ulaşmadığını bir sayaç ile denetleyerek gerçekleştirilen çevrim yapıları (for döngüsü)**
- ✓ **Çevrimin sona ermesini bir koşula bağlı olarak kontrol eden çevrim yapıları (while döngüsü)**



DÖNGÜLER (devam)

for Döngüsü

Bir **for** (için) döngünün genel formatı

```
for döngü değişkeni=ifade  
    deyimler
```

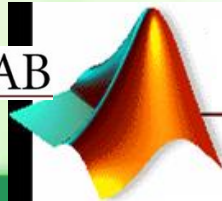
```
end
```

while Döngüsü

Bir **while** (süresince veya iken) döngüsünün genel formatı

```
while ifade  
    deyimler
```

```
end
```



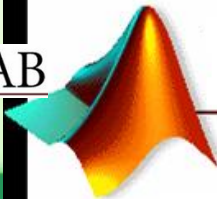
for DÖNGÜSÜ

```
for döngü değişkeni = başlangıç:artış miktarı:bitiş  
    komutlar  
end
```

```
for döngü değişkeni = başlangıç:bitiş  
    komutlar  
end
```

artış miktarı otomatik olarak 1 atanır.

MATLAB



Örnek: 1 ile klavyeden girilen herhangi bir sayı arasındaki sayıların toplamını ve çarpımını hesaplayarak ekrana basan bir MATLAB programı yazınız.

```
clc;
```

```
clear;
```

```
n=input('lutfen bir sayi giriniz= ');
```

```
toplam=0; % toplamada etkisiz eleman
```

```
carpim=1; % carpimada etkisiz eleman
```

```
for i=1:n
```

```
    toplam=toplam+i ;
```

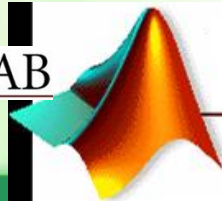
```
    carpim=carpim*i ;
```

```
end
```

```
toplam % Gauss Teoremi
```

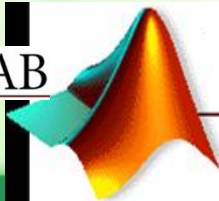
```
carpim % n sayisinin faktoriyeli
```

MATLAB



UYGULAMA

```
fprintf('for dongusu testi \n');  
for i = 4:-1:1  
    fprintf('for dongusu icindeyiz: i = %d \n',i);  
end  
fprintf('\nfor dongusunun sonu \n');
```



while DÖNGÜSÜ

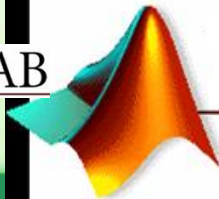
while şart

ifade_1

ifade_2

ifade_n

end



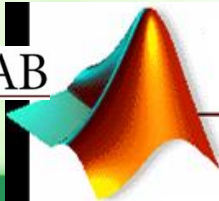
UYGULAMA

Ekrana adınızı ve soyadınızı 10 kez yazan bir MATLAB programını **while** döngüsü kullanarak oluşturunuz.

```
clc;  
clear;  
  
a=1;  
while a<=10  
    disp('Deniz Dal');  
    a=a+1;  
end
```

for Döngüsü ile Çözüm

```
clc;  
clear;  
  
for i=1:10  
    disp('Deniz Dal');  
end
```



Uygulama

Asağıdaki **while** döngüsü kaç kere işletilir?

```
devamEt=1;
```

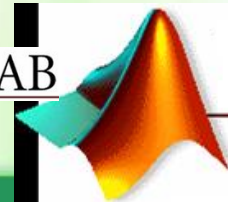
```
a=0;
```

```
while devamEt  
    disp('Deniz Dal');  
    a=a+1;
```

```
end
```

CTRL+C

Yandaki programı ekrana 10 kere
“Deniz Dal” yazacak şekilde nasıl
modifiye edersiniz??????



break DEYİMİ

for veya **while** döngülerinde program akışını kontrol edebilmenin diğer bir yolu da **break** deyimini kullanmaktır. **break** deyimini döngü gövdesi içerisinde kullanmak, döngünün durmasını ve döngüden sonra gelen ilk ifade veya komutun işletilmesini sağlar.

Örnek:

```
clc;clear;
```

```
for i=1:10
```

```
    if i==4
```

```
        break;
```

```
    end
```

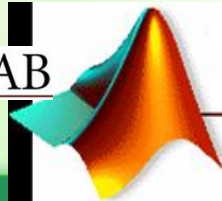
```
    fprintf('i= %d \n', i);
```

```
end
```

```
disp('dongu break deyimi ile sonlandırıldı.');
```

EKRAN ÇIKTISI ???

MATLAB



continue DEYİMİ

SADECE for döngülerinde, program akışı ayrıca **continue** deyimi ile kontrol edilir. **continue** deyimi işletildiğinde sayaç değişkeni bir sonraki değerine artırılır, **continue** deyiminden sonraki bütün herşey ihmal edilir ve program for döngüsünün ilk deyiminden itibaren tekrar çalışmaya başlar.

Örnek:

```
clc;clear;
```

```
for i=1:10
```

```
    if i==4
```

```
        continue;
```

```
    end
```

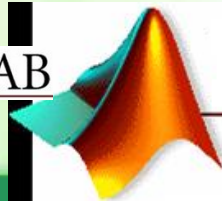
```
    fprintf('i= %d \n', i);
```

```
end
```

```
disp('dongunun isleyisi continue deyimi ile degistirildi.');
```

EKRAN ÇIKTISI ???

MATLAB



BİR SATIR VEKTÖRÜNÜ BİLGİ DEPOLAMAK İÇİN KULLANMA

$A=[]$; %Baslangicta Bos

1. METOD :

$A=[A \ x]$; % $A=[x]$

$A=[A \ y]$; % $A=[x \ y]$

.

.

2. METOD :

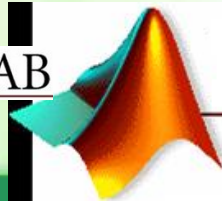
$A(1)=x$; % $A=[x]$

$A(2)=y$; % $A=[x \ y]$

.

.

Başlangıçta boş olan bir satır vektörünün içine sırasıyla x ve y elemanları ekleniyor.

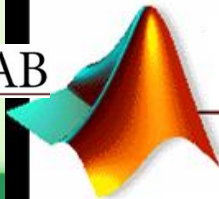


İççe for Döngüleri

```
clc;
clear;
carpim=[ ]; % Bos Vektor
for i = 1:5
    toplam = 0;
    for j = 1:5
        toplam = toplam + j;
    end
    x = toplam * i;
    carpim=[carpim x];
end
carpim
```

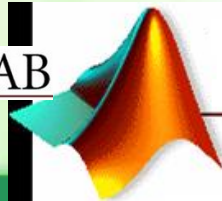
EKRAN ÇIKTISI ???

MATLAB



Örnek: Çarpım tablosunu ekrana basan bir MATLAB programı yazınız.

```
clc;  
clear;  
for a=1:10  
    for b=1:10  
        carpim=a*b;  
        fprintf('%d * %d = %d \n', a, b, carpim);  
    end  
end
```



ÖRNEK

Yarıçap değerleri 1,2,3,4,5 için bir kürenin hacmini ($H=4/3\pi r^3$) hesaplayan bir MATLAB programı yazınız.

elemanter yöntem:

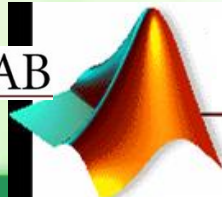
```
r=1:5;  
hacim=(4/3)*pi*r.^3;  
disp([r;hacim])
```

for yöntemi:

```
for r=1:5  
    hacim=(4/3)*pi*r^3;  
    disp([r,hacim])  
end
```

while yöntemi:

```
r=1;  
while r<=5  
    hacim=(4/3)*pi*r^3;  
    disp([r,hacim])  
    r=r+1;  
end
```



Örnek: Girilen ay'a göre gün sayısını hesaplayan MATLAB programı.

```
clc;  
clear;
```

```
ay=input('Hangi ayin gun sayisini ogrenmek istiyorsunuz (1-12)= ');
```

```
if ay==1 | ay==3 | ay ==5 | ay==7 | ay==8 | ay==10 | ay==12
```

```
    disp('Bu Ay 31 gunden olusur.');
```

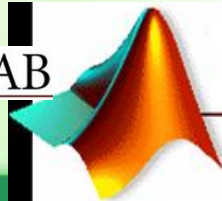
```
elseif ay==2
```

```
    disp('Bu Ay 28 gunden olusur.');
```

```
else
```

```
    disp('Bu Ay 30 gunden olusur.');
```

```
end
```



switch Şartlı Deyimi

```
switch(durum)
```

```
    case{durum1}
```

```
        işlemler
```

```
    case{durum2}
```

```
        işlemler
```

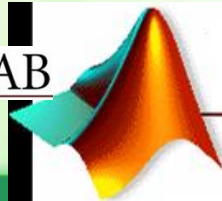
```
    .
```

```
    .
```

```
    otherwise
```

```
        % isteğe bağlı
```

```
end
```



Örnek: 1 ile 10 arasında girilen bir sayının tek ya da çift olduğunu ekrana basan bir MATLAB programını **switch** deyimini kullanarak yazınız.

```
clc;clear;
```

```
sayi=input('1 ile 10 arasinda bir sayi giriniz= ');
```

```
switch(sayi)
```

```
    case{1,3,5,7,9}
```

```
        disp('Bu sayi tektir.');
```

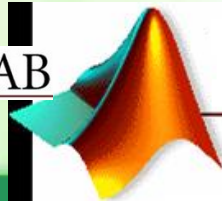
```
    case{2,4,6,8,10}
```

```
        disp('Bu sayi cifttir.');
```

```
    otherwise
```

```
        disp('Bu sayi 1 ile 10 araliginin disinda.');
```

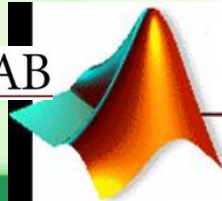
```
end
```



UYGULAMA

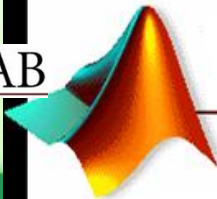
Girilen ay numarasına göre, o ayın gün sayısını veren MATLAB programını **switch** yapısını kullanarak yazınız.

```
clc;clear;
ay = input('Bir ay numarası giriniz (1-12)= ');
switch(ay)
    case{1,3,5,7,8,10,12}
        disp('Bu Ay 31 günden oluşur. ');
    case{2}
        disp('Bu Ay 28 günden oluşur. ');
    case{4,6,9,11}
        disp('Bu Ay 30 günden oluşur. ');
    otherwise
        disp('Yanlış bir ay no girdiniz. ');
end
```



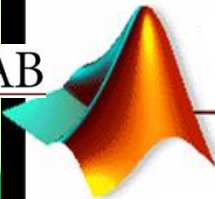
HATA AYIKLAMA (DEBUGGING)

M dosyası editörünün hata ayıklamaya (**debugging**) yarayan fonksiyonlarını ve çalışma alanı (**workspace**) penceresini kullanarak çalışan programınızdaki **(!! BEKLEDİĞİNİZ GİBİ ÇALIŞMAYAN !!)** hataları bulabilirsiniz. Bunun için herhangi bir program satırına kesme noktası (**breakpoint**) koymalı ve kodunuzu satır satır işletmelisiniz. (Derste bu konuyu özetleyen canlı bir uygulama yapılacaktır.)



UYGULAMA

İçerisinde rastgele oluşturulmuş tam sayılarla dolu (1 ile 50 arasında) 20 elemanlı bir satır vektörünün **en küçük** ve **en büyük** elemanını bularak ekrana basan bir MATLAB programı yazınız. (**min ve max** fonksiyonlarını kullanmıyoruz. Kendi **min ve max** fonksiyonlarımızı yazmaya çalışıyoruz.)



ÇÖZÜM

```
clc;clear;
A=round(1+49*rand(1,20));
%Dizinin En Kucuk ve En Buyuk Elemaninin Dizinin
%ilk Elemani Oldugunu Farzet
enKucuk=A(1);
enBuyuk=A(1);
for i=2:length(A)
    if A(i)<enKucuk
        enKucuk=A(i);
    end
    if A(i)>enBuyuk
        enBuyuk=A(i);
    end
end
A
disp(['Bu Dizinin En Kucuk Elemani : ' num2str(enKucuk)]);
disp(['Bu Dizinin En Buyuk Elemani : ' num2str(enBuyuk)]);
```

