



# PROGRAMLAMA TEMELLERİ

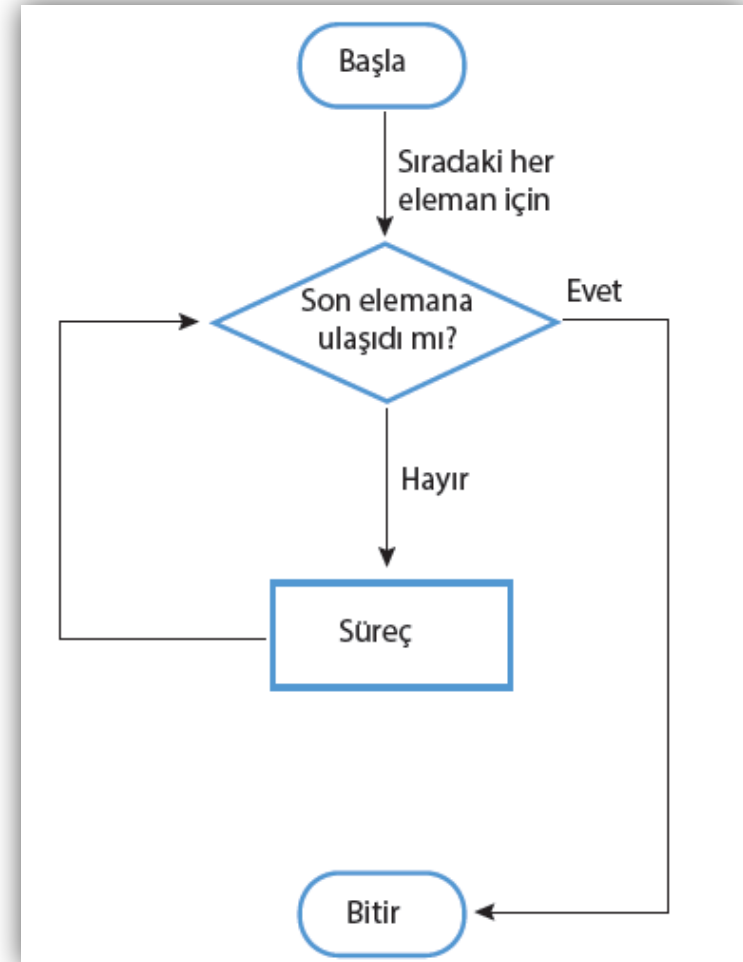
Öğr. Gör. Erhan AKAGÜNDÜZ

# DÖNGÜLER

- ❑ Programlama dillerinde karar yapıları gibi sık kullanılan başka bir yapı da döngülerdir.
- ❑ Program içinde kod bloklarının istenen sayıda tekrar etmesini sağlayan yapılara **döngü** adı verilir.
- ❑ Python programlama dilinde **for** ve **while** döngüleri bulunmaktadır.

# FOR DÖNGÜSÜ

- ❑ Şart doğru olduğu sürece işlemlerin tekrarını sağlayan döngü yapısıdır.
- ❑ For döngüsü belirli bir şart sağlanana kadar belirlenen kod bloklarını tekrarlar.
- ❑ For döngüsünün yapısı yandaki gibidir:



# FOR DÖNGÜSÜ - RANGE KULLANIMI

- ❑ Döngünün başlangıç ve bitiş değeri belli olan durumlarda kullanılan fonksiyondur.
- ❑ Varsayılan olarak 0'dan (sıfır) başlayarak birer birer artar.
- ❑ Range fonksiyonunun bitiş değeri döngü dışında kabul edilir.
- ❑ Yandaki örnekte bir başlangıç değeri verilmediği için döngü **0'dan (sıfır) başlar** ve 5'e kadar devam eder (5 hariç).
- ❑ **sayilar** ismiyle oluşturulan değişken **print** fonksiyonu ile ekrana yazdırılmıştır.

```
for sayilar in range(5):  
    print(sayilar)
```

```
0  
1  
2  
3  
4
```

# FOR DÖNGÜSÜ - RANGE KULLANIMI

- ❑ Aşağıdaki örnekte başlangıç ve bitiş değerleri birlikte verilmiştir.
- ❑ Ekran çıktısına bakıldığında başlangıç değeri olan **5**'ten başlayarak bitiş değerine kadar olan sayılar (**bitiş değeri dâhil değil**) ekrana yazdırılmıştır.

```
for sayilar in range(5,10):  
    print(sayilar)
```

5  
6  
7  
8  
9

# FOR DÖNGÜSÜ - RANGE KULLANIMI

- ❑ Bu örnekte başlangıç ve bitiş değerleri ile birlikte artış değeri de verilmiştir.
- ❑ Başka bir ifadeyle döngünün 5'ten başlayarak 20'ye kadar 3'er 3'er artması sağlanmıştır.
- ❑ Burada yine dikkat edilmesi gereken nokta 17 sayısından sonra 20 sayısının son değer olduğu için çıktıda görülmemesidir.

```
for sayilar in range(5,20,3):  
    print(sayilar)
```

5

8

11

14

17

# FOR DÖNGÜSÜ - RANGE KULLANIMI

- ❑ Bu örnekte 20'den başlayarak 5'e kadar (5 dâhil değil) 3'er azalan sırada sayılar ekrana yazdırılmıştır.

```
for sayilar in range(20,5,-3):  
    print(sayilar)
```

```
20  
17  
14  
11  
8
```

# FOR DÖNGÜSÜ - RANGE KULLANIMI

- ❑ Range fonksiyonu ile else anahtar sözcüğü de kullanılarak döngü sonunda bir mesaj verilebilir.
- ❑ Kullanımı şu şekildedir:

```
for sayilar in range(5,20,3):  
    print(sayilar)  
else:  
    print("Döngü bitti")
```

```
5  
8  
11  
14  
17  
Döngü bitti
```



# FOR DÖNGÜSÜ - RANGE KULLANIMI

❏ **Örnek:** for döngüsü ile 1'den 10'a kadar olan sayıların toplamını bularak ekrana yazdırınız.

```
toplam = 0
for sayilar in range(11):
    toplam = toplam + sayilar
print("Sayıların toplamı: ", toplam)

Sayıların toplamı:  55
```

# FOR DÖNGÜSÜ - IN KULLANIMI

- ❑ **in** operatörü bir elemanın listede olup olmadığını kontrol eder.
- ❑ in operatörü metinsel (string) ifadeleri de harf harf ekrana yazdırabilir.
- ❑ **Kullanımı şu şekildedir:**

```
for harfler in "Döngü":  
    print(harfler)
```

D  
ö  
n  
g  
ü

# FOR DÖNGÜSÜ - IN KULLANIMI

- ❑ **in** operatörü metinsel (**string**) ifadeleri de harf harf ekrana yazdırabileceği gibi bunların istenilen sayıda tekrar ile yan yana yazabiliriz.
- ❑ **Kullanımı şu şekildedir:**

```
for harfler in "PYTHON":  
    print(harfler*10)
```

```
PPPPPPPPPP  
YYYYYYYYYY  
TTTTTTTTTT  
HHHHHHHHHH  
OOOOOOOOOO  
NNNNNNNNNN
```

# FOR DÖNGÜSÜ - IN KULLANIMI

- ❑ **Örnek:** 10-20 arası sayılardan oluşan **sayilar** isimli bir liste oluşturarak liste içinde 3'e tam bölünen sayıları ekrana yazdırınız.

```
sayilar=[10,11,12,13,14,15,16,17,18,19,20]  
for sayi in sayilar:  
    if sayi%3==0:  
        print(sayi)
```

```
12  
15  
18
```

# FOR DÖNGÜSÜ - IN KULLANIMI

- ❑ **Örnek:** **alan\_adi** isimli, değeri **bilişim** olan bir değişken tanımlayarak içinde kaç adet **"i"** harfi olduğunu bulup ekrana yazdırınız.

```
alan_adi = "bilışim"
toplam = 0
for aranan in alan_adi:
    if aranan == "i":
        toplam = toplam + 1
print("Bu kelimedede toplam",toplam,"adet 'i' vardır")

Bu kelimedede toplam 3 adet 'i' vardır
```

# FOR DÖNGÜSÜ - IN KULLANIMI

❑ **Örnek:** Kullanıcıdan bir ifade ve aranacak harf girmesini isteyiniz. Girilen ifadede kaç tane “a” harfi olduğunu bularak ekrana yazdırınız.

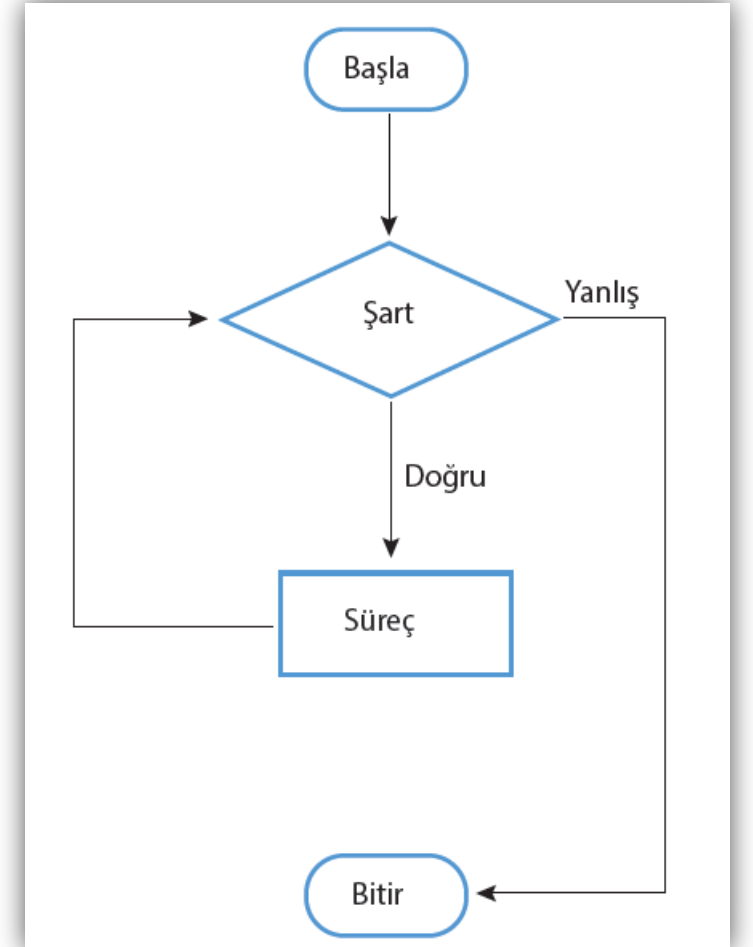
❑ **Öğrenciler çözsün!**

```
kelime = input("Bir kelime yazınız: ")
harf = input("Girdiğiniz kelimedede hangi harf aransın: ")
toplam = 0
for aranan in kelime:
    if aranan == harf:
        toplam = toplam + 1
print("Bu kelimedede toplam",toplam,"adet", harf,"vardır")
```

```
Bir kelime yazınız: ankara
Girdiğiniz kelimedede hangi harf aransın: a
Bu kelimedede toplam 3 adet a vardır
```

# WHILE DÖNGÜSÜ

- ❑ **While** döngüsü hemen hemen tüm programlama dillerinde bulunmaktadır.
- ❑ Test edilen ifade doğru (true) olduğu sürece kodları tekrarlamaktadır.
- ❑ While yapısı genellikle kod bloğunun kaç kez tekrar edileceğinin bilinmediği durumlarda kullanılmaktadır.
- ❑ While döngüsünün yapısı yandaki gibidir:



# WHILE DÖNGÜSÜ

- ❑ Bu örnekte **i** değişkenine ilk değer olarak 0 (sıfır) atanmıştır.
- ❑ **while (i<5):** satırı ile **i** değeri 5 olana kadar (5 dâhil değil) döngü devam eder.
- ❑ Her adımda bir kez “Kodlama” ifadesi ekrana yazdırılır ve **i** değeri **1** arttırılır.
- ❑ **i** değeri sırasıyla 0,1,2,3 ve 4 olur.
- ❑ Yani döngü **5 kez döner** ve program sonlanır.

```
i=0
while (i<5):
    print("Kodlama")
    i=i+1
```

```
Kodlama
Kodlama
Kodlama
Kodlama
Kodlama
```



# WHILE DÖNGÜSÜ

- ❑ **Sonsuz döngü:** Programlama dillerinde döngü oluştururken yapılacak bir mantık hatası sonsuz döngüye neden olabilir.
- ❑ Sonsuz döngüde program sürekli çalışacaktır.
- ❑ Sonsuz döngüden çıkmak için **Ctrl+C** tuş kombinasyonu kullanılabilir.

```
i=15
while (i<20):
    print(i)
    i=i-1
```

# WHILE DÖNGÜSÜ

❑ **Örnek:** Klavyeden girilen sayının faktöriyelini hesaplayan programı while ile yazınız.

```
i=1
sonuc=1
faktoriyel=int(input("Faktoriyeli hesaplanacak sayıyı giriniz: "))
while (i<=faktoriyel):
    sonuc=i*sonuc
    i=i+1
print("Sonuç=",sonuc)
```

```
Faktoriyeli hesaplanacak sayıyı giriniz: 5
Sonuç= 120
```

# WHILE DÖNGÜSÜ

- ❏ **Örnek:** Girilen sayı 0 (sıfır) olana kadar girilen tüm sayıları toplayan ve ekranda gösteren programı yazınız

```
toplam = 0
sayi = 1
while (sayi!=0):
    sayi = int(input("Bir sayı giriniz: "))
    toplam = toplam + sayi
print("Sonuç",toplam)
```

```
Bir sayı giriniz: 2
Bir sayı giriniz: 4
Bir sayı giriniz: 6
Bir sayı giriniz: 0
Sonuç 12
```

# WHILE DÖNGÜSÜ

❑ **Örnek:** Çarpım tablosunu iç içe döngü kurarak kodlayınız.

```
for i in range(1,11):  
    for j in range(1,11):  
        print(i,"*",j,"=",i*j)  
    print("=====")
```

```
1 * 1 = 1  
1 * 2 = 2  
1 * 3 = 3  
1 * 4 = 4  
1 * 5 = 5  
1 * 6 = 6  
1 * 7 = 7  
1 * 8 = 8  
1 * 9 = 9  
1 * 10 = 10  
=====  
2 * 1 = 2  
2 * 2 = 4  
2 * 3 = 6  
2 * 4 = 8  
2 * 5 = 10  
2 * 6 = 12  
2 * 7 = 14  
2 * 8 = 16  
2 * 9 = 18  
2 * 10 = 20  
=====  
3 * 1 = 3  
3 * 2 = 6  
3 * 3 = 9  
3 * 4 = 12  
3 * 5 = 15  
=====
```

# BREAK VE CONTINUE DEYİMLERİ

- ❑ Break komutu döngüleri sonlandırır.
- ❑ Programlamada öngörülemeyen bir durum gerçekleştiğinde break komutu döngüden çıkılmasına imkân tanır.
- ❑ Döngüden çıkıldıktan sonra ise döngü sonrasındaki ilk satırdan kod çalıştırılmaya devam eder.

# BREAK VE CONTINUE DEYİMLERİ

- ❑ **Örnek:** sayılar listesindeki çift sayıları atlayarak ve tek sayıları ekran çıktısında listeleyiniz.

```
sayilar=[20,23,79,88,111,65,36]  
for sayi in sayilar:  
    if sayi%2 == 0:  
        continue  
    print(sayi)
```

```
23  
79  
111  
65
```

# BREAK VE CONTINUE DEYİMLERİ

- ❑ Bu örnekte kullanılan **random** komutu ile 1-20 arasında rastgele bir sayının hafızaya alınması sağlanmıştır.
- ❑ Seçilen bu rastgele sayı çift bir sayı olana kadar döngü devam eder.
- ❑ Çift sayı tutulduğunda ise döngü sona erer.

```
import random
```

```
while True:
```

```
    n = random.randint(1,20)
```

```
    print("Rasgele seçilen", n)
```

```
    if n%2==0:
```

```
        print("Çift sayı seçildi. Döngü bitti!")
```

```
        break
```

```
Rasgele seçilen 13
```

```
Rasgele seçilen 7
```

```
Rasgele seçilen 7
```

```
Rasgele seçilen 3
```

```
Rasgele seçilen 12
```

```
Çift sayı seçildi. Döngü bitti!
```