

Döngüler

2025

Döngüler (loop)

- Döngü, bir işi, belirlediğiniz sayıda (veya koşulda) yapan kod bloklarıdır.
- Tekrar eden işlemlerde kullanılır.
- Döngüler koşul doğru olduğu sürece dönmektedir.
- Koşul sağlanmadığı taktirde döngüden çıkmaktadır.



While Döngüsü

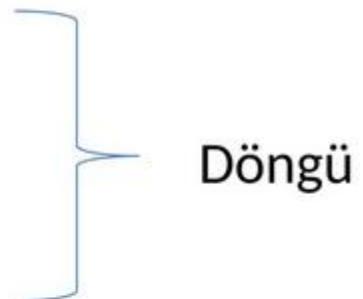
- ... iken,
- ... olduğu sürece,
anımlarına gelmektedir.
- Koşul sağlandığı sürece döngü devam eder.

Döngüler (loop)

while

>>> **while** koşul :
uygulanacak komutlar bloğu

(koşul sağlandığı sürece)



Döngü

Bloğun dışında komutlar..

Döngüler (loop)

i=1

while i<=5 :

 print(i)

 i += 1



1

2

3

4

5

Değişkenimiz : i

Koşulumuz : i<=5

Sayacımız : i += 1

Sonuç:

1 den 5 kadar yazan döngü

Alıştırmalar

```
sayac = 1  
while (sayac < 10) :  
    print ("Bu sayacımız:", sayac)  
    # sayac = sayac + 1  
    sayac += 1  
  
print ("Hoşçakal !")
```

1 ' den 100 ' e kadar olan aralıktaki çift sayıları while döngüsü kullanarak yazınız.

Lütfen deneyiniz..

```
sayı=0  
while sayı < 100 :  
    sayı += 1  
    if sayı % 2 == 0 :  
        print(sayı)  
    # if bloğundan çıkıyoruz.  
#while bloğundan çıkıyoruz.
```

For Döngüsü

- For -> İçin anlamına geliyor..
- Genelde in ifadesi ile kullanılmaktadır.
- İn -> İçinde anlamına geliyor..
- İşlemlerin tekrar sayısının önceden belli olduğu durumlarda kullanılır.
- Tekrarlama veya yineleme anımlarına gelen iterasyon ile birlikte kullanılır.

Döngüler (loop)

for

>>> **for** değişken **in** sıralama :
ifadeler (komutlar)

Bloğun dışında kalan komutlar..

Döngüler (loop)

```
for karakter in "Python":  
    print ("Harf:", karakter)
```

Harf: P
Harf: y
Harf: t
Harf: h
Harf: o
Harf: n

```
# 0,1,2,3,4  
for x in range(5):  
    print(x)
```

0	sayi = 1	1
1	for sayi in range(1,11):	2
2	print(sayi)	3
3		4
4		5
		6
		7
		8
		9
		10

Döngüler (loop)

range()

- Ardışık sayılarından oluşan iterasyon edilebilen bir nesnedir.

range(başlangıç, bitiş, basamak)

Varsayılanlar;

Başlangıç: 0

Basamak: 1

Döngüler (loop)

for in range()

```
for x in range(0,24,3):  
    print(x)
```

0
3
6
9
12
15
18
21

Döngüler (loop)

Break ve Continue

- **Break**, devam eden bir süreci kesintiye uğratmak için.
- **Continue**, kendisinden sonra gelen her şeyin es geçilip döngünün başına dönülmesini sağlamak için.
- Genellikle while, for veya if ile kullanılmaktadır..

Döngüler (loop)

Break Örneği

while True:

```
    sehir = input("Lütfen bir şehir yazınız:")
```

```
    if len(sehir) < 3:
```

```
        print("Şehir 3 karakterden az olmamalı!")
```

```
    else:
```

```
        print("Şehir belirlendi!")
```

```
        break
```



Döngüler (loop)

Continue Örneği

while True:

s = input("Bir sayı giriniz: ")

print("Sayınız: ",s)

if s == "x": # x girilince program sona erecek.

break

if len(s) <= 3:

 continue # 3 haneden az olursa başa dön

 print("En fazla üç haneli bir sayı girebilirsiniz.")

while döngüsü: Koşul sağlandığı sürece çalışır

for döngüsü: Belirli bir dizi üzerinde iterasyon yapar

break: Döngüden tamamen çıkar

continue: Döngünün geri kalanını atlar, başa döner

Alıştırmalar

Şifre Doğrulama Sistemi (while + break)

Soru: Kullanıcıya maksimum 3 deneme hakkı veren bir şifre doğrulama programı yazın.
Doğu şifre girildiğinde döngüden çıkışlsın.

```
python

print("==> ŞİFRE DOĞRULAMA SİSTEMİ ==>")
print()

dogru_sifre = "Python123"
deneme_hakki = 3
deneme_sayisi = 0

while deneme_sayisi < deneme_hakki:
    sifre = input(f"Şifrenizi giriniz (Kalan hak: {deneme_hakki - deneme_sayisi}): ")
    deneme_sayisi += 1

    if sifre == dogru_sifre:
        print("\n✓ Giriş başarılı! Hoş geldiniz.")
        break # Doğru şifre girildi, döngüden çıkış
    else:
        kalan = deneme_hakki - deneme_sayisi
        if kalan > 0:
            print(f"\n✗ Yanlış şifre! {kalan} deneme hakkınız kaldı.\n")
        else:
            print("\n✗ Deneme hakkınız bitti! Hesabınız kilitlendi.")
```

Şifre Doğrulama Sistemi (while + break)

Örnek Çalıştırma:

```
==== ŞİFRE DOĞRULAMA SİSTEMİ ===
```

```
Şifrenizi giriniz (Kalan hak: 3): python
✗ Yanlış şifre! 2 deneme hakkınız kaldı.
```

```
Şifrenizi giriniz (Kalan hak: 2): Python
✗ Yanlış şifre! 1 deneme hakkınız kaldı.
```

```
Şifrenizi giriniz (Kalan hak: 1): Python123
```

```
✓ Giriş başarılı! Hoş geldiniz.
```

Asal Sayı Bulucu (for + continue + break)

Soru: Kullanıcının girdiği aralıktaki asal sayıları bulan program yazın. Asal olmayan sayıları atlamak için continue kullanın.

```
python

print("== ASAL SAYI BULUCU ==")
print()

baslangic = int(input("Başlangıç sayısı: "))
bitis = int(input("Bitiş sayısı: "))

print(f"\n{baslangic} ile {bitis} arasındaki asal sayılar:")
print("-" * 40)

asal_sayilar = []
bulunan_adet = 0

for sayı in range(baslangic, bitis + 1):
    # 1 ve negatif sayıları atla
    if sayı < 2:
        continue

    # Asal kontrolü
    asal_mi = True
```

```
for bölen in range(2, int(sayı ** 0.5) + 1):
    if sayı % bölen == 0:
        asal_mi = False
        break # Bölüm bulundu, kontrol etmeye gerek yok

if asal_mi:
    asal_sayilar.append(sayı)
    bulunan_adet += 1
    print(f"{sayı}", end=" ")

    # Her 10 sayıda bir satır atla
    if bulunan_adet % 10 == 0:
        print()

print("\n" + "-" * 40)
print(f"Toplam {bulunan_adet} asal sayı bulundu.")
print(f"İlk asal: {asal_sayilar[0]} if asal_sayilar else 'Yok'")
print(f"Son asal: {asal_sayilar[-1]} if asal_sayilar else 'Yok'")
```

```

# Asal Sayı Bulucu (for + continue + break)

```
Örnek Çalıştırma:
```

```
...
```

```
==== ASAL SAYI BULUCU ===
```

```
Başlangıç sayısı: 10
```

```
Bitiş sayısı: 50
```

```
10 ile 50 arasındaki asal sayılar:
```

```

11 13 17 19 23 29 31 37 41 43
```

```
47

```

```
Toplam 11 asal sayı bulundu.
```

```
İlk asal: 11
```

```
Son asal: 47
```

# Öğrenci Not Ortalaması Hesaplama (w + b + c)

```
python

print("== ÖĞRENCİ NOT SİSTEMİ ==")
print("Not girmek için sayı, çıkmak için 'q' yazın")
print("-" * 50)

notlar = []
toplam = 0
gecersiz_giris = 0

while True:
 giriş = input(f"\n{len(notlar) + 1}. notu giriniz (0-100 arası, çıkış: q): ")

 # Çıkış kontrolü
 if giriş.lower() == 'q':
 print("\n✓ Not girişi sonlandırıldı.")
 break

 # Sayısal değer kontrolü
 try:
 not_degeri = float(giriş)
 except ValueError:
 print("⚠ Geçersiz giriş! Lütfen sayı giriniz.")
 gecersiz_giris += 1
 continue # Geçersiz giriş, döngünün başına dön

 # Not aralığı kontrolü
 if not_degeri < 0 or not_degeri > 100:
 print("⚠ Not 0-100 arasında olmalıdır!")
 gecersiz_giris += 1
 continue # Geçersiz not, döngünün başına dön

 # Geçerli not ekleme
 notlar.append(not_degeri)
 toplam += not_degeri
 print(f"✓ Not eklendi: {not_degeri}")

 # Sonuçları göster
 print("\n" + "=" * 50)
 print(" SONUÇLAR")
 print("=" * 50)

 if len(notlar) > 0:
 ortalama = toplam / len(notlar)

 print(f"Girilen not sayısı : {len(notlar)}")
 print(f"Geçersiz giriş : {gecersiz_giris}")
 print(f"En yüksek not : {max(notlar):.2f}")
 print(f"En düşük not : {min(notlar):.2f}")
 print(f"Not ortalaması : {ortalama:.2f}")
```

# Öğrenci Not Ortalaması Hesaplama (w + b + c)

```
Harf notu belirleme
if ortalama >= 90:
 harf = "AA"
elif ortalama >= 85:
 harf = "BA"
elif ortalama >= 80:
 harf = "BB"
elif ortalama >= 75:
 harf = "CB"
elif ortalama >= 70:
 harf = "CC"
elif ortalama >= 65:
 harf = "DC"
elif ortalama >= 60:
 harf = "DD"
elif ortalama >= 50:
 harf = "FD"
else:
 harf = "FF"

print(f"Harf notu : {harf}")

if ortalama >= 60:
 print("\n🎉 Tebrikler! Dersten geçtiniz.")
else:
 print("\n🔴 Dersten kaldınız. Daha fazla çalışmalısınız.")
```

```
=====
SONUÇLAR
=====
Girilen not sayısı : 3
Geçersiz giriş : 1
En yüksek not : 92.00
En düşük not : 78.00
Not ortalaması : 85.00
Harf notu : BA

🎉 Tebrikler! Dersten geçtiniz.
=====
```

```
else:
 print("⚠️ Hiç not girilmemi!")

print("=" * 50)
```

```

Örnek Çalıştırma:
```

== ÖĞRENCİ NOT SİSTEMİ ==  
Not girmek için sayı, çıkmak için 'q' yazın  
-----

1. notu giriniz (0-100 arası, çıkış: q): 85  
✓ Not eklendi: 85.0

2. notu giriniz (0-100 arası, çıkış: q): 92  
✓ Not eklendi: 92.0

3. notu giriniz (0-100 arası, çıkış: q): 150  
⚠️ Not 0-100 arasında olmalıdır!

4. notu giriniz (0-100 arası, çıkış: q): 78  
✓ Not eklendi: 78.0

5. notu giriniz (0-100 arası, çıkış: q): q  
✓ Not girişi sonlandırdı.

# Öğrenci Not Ortalaması Hesaplama ( $w + b + c$ )

---