

MUDANYA
UNIVERSITY



Döngüler



- Döngü: Tekrar tekrar çalıştırılan bir dizi komut
- Bir programın bir bölümünün belirli bir hedefe ulaşılana kadar defalarca tekrarlanması
- **While döngüsü**: Koşul sağlandığı sürece while deyimi içindeki ifadeler çalıştırılır
- Genellikle iterasyon sayısı önceden bilinmediğinde while döngüleri kullanılır

while koşul :
→ kod bloğu

While Döngüsü

4

Döngü	Çıktı	Açıklama
<pre>i = 0 toplam = 0 while toplam < 10 : i = i + 1 toplam = toplam + i print(i, toplam)</pre>	<pre>1 1 2 3 3 6 4 10</pre>	Toplam değeri 10 olduğunda, döngü koşulu yanlış olur ve döngü sona erer.
<pre>i = 0 toplam = 0 while toplam < 10 : i = i + 1 toplam = toplam - i print(i, toplam)</pre>	<pre>1 -1 2 -3 3 -6 4 -10 ...</pre>	Toplam hiçbir zaman 10'a ulaşmadığından, bu bir “ sonsuz döngü ”dür
<pre>i = 0 toplam = 0 while toplam < 0 : i = i + 1 toplam = toplam - i print(i, toplam)</pre>	(Çıktı yok)	Koşul ilk kez kontrol edildiğinde toplam < 0 ifadesi yanlıştır ve döngü hiçbir zaman yürütülmez .

Yaygın Kullanılan While Döngüsü Algoritmaları

- Bir dizi girdinin toplamının hesaplanması

```
toplam = 0.0
girdi_str = input("bir deger girin: ")
while girdi_str != "":
    girdi = float(girdi_str)
    toplam = toplam + girdi
    girdi_str = input("bir deger girin: ")
print(toplam)
```

Yaygın Kullanılan While Döngüsü Algoritmaları

- Bir dizi girdinin ortalamasının hesaplanması
 - Kaç tane değer olduğunu sayın ve toplamı bu sayıya bölün

```
toplam = 0.0
sayac = 0
girdi_str = input("bir deger girin: ")
while girdi_str != "":
    sayi = float(girdi_str)
    toplam = toplam + sayi
    sayac = sayac + 1
    girdi_str = input("bir deger girin: ")
if sayac > 0:
    ortalama = toplam / sayac
else:
    ortalama = 0.0
print(ortalama)
```

Yaygın Kullanılan While Döngüsü Algoritmaları

- Eşleşmeleri sayma
 - Genellikle belirli bir koşulu kaç değerin karşıladığını bilmek için kullanılır

```
negatif_sayisi = 0
girdi_str = input("bir deger girin: ")
while girdi_str != "":
    sayi = int(girdi_str)
    if sayi < 0:
        negatif_sayisi += 1
    girdi_str = input("bir deger girin: ")
print(f"{negatif_sayisi} tane negatif sayi girdiniz")
```

Yaygın Kullanılan While Döngüsü Algoritmaları

- Geçerli bir eşleşme bulunana kadar kullanıcıdan veri isteme
 - Programı sonlandırmak yerine, geçerli bir değer sağlanana kadar kullanıcıdan verileri girmesini istemeye devam edilir

```
kontrol = False
while not kontrol :
    deger = int(input("100'den küçük pozitif bir tamsayı girin: "))
    if deger > 0 and deger < 100 :
        kontrol = True
    else :
        print("Gecersiz giris")
```

"while not valid" ifadesi kontrol değeri True olana kadar döngünün çalışmasını sağlar

Girilen değer 0 ile 100 arasında değilse kullanıcıya "Gecersiz giris" mesajı verilir ve kullanıcı geçerli bir değer girene kadar döngü devam eder.

Yaygın Kullanılan While Döngüsü Algoritmaları

- Maksimum ve minimum değeri bulma

```
maksimum = 0
girdi_str = input("bir deger girin: ")
while girdi_str != "":
    sayi = float(girdi_str)
    if sayi > maksimum:
        maksimum = sayi
    girdi_str = input("bir deger girin: ")
print(maksimum)
```

```
minimum = 1000
girdi_str = input("bir deger girin: ")
while girdi_str != "":
    sayi = float(girdi_str)
    if sayi < minimum:
        minimum = sayi
    girdi_str = input("bir deger girin: ")
print(minimum)
```


- **For döngüsü**: Bir öge koleksiyonu içeren veya depolayan bir nesne olan herhangi bir «container» içeriği üzerinde yineleme yapmak için kullanılır
- Container: Nesnelerden oluşan bir koleksiyonu tutabilen ve koleksiyonu yönetmek ve koleksiyona erişmek için bir mekanizma sağlayan liste gibi bir veri yapısı

for degisken *in* container :

→ kod bloğu

```
isimler = ["asli", "yesim", "ufuk", "deniz", "onurkan"]  
for i in isimler:  
    print(i)
```

For Döngüsü

11

- **Range** fonksiyonu: Bir dizi tamsayı üretir
 - range(başlangıç, bitiş, adım büyüklüğü)
 - Adım büyüklüğünün varsayılan değeri 1'dir

for degisken *in* range() :

→ kod bloğu

```
for i in range(1, 10): # i = 1, 2, 3, ..., 9  
    print(i)
```

=

```
i = 1  
while i < 10 :  
    print(i)  
    i = i + 1
```

Döngü	i değerleri	Açıklama
for i in range(6) :	0, 1, 2, 3, 4, 5	Döngü 6 kez çalıştırılır.
for i in range(10, 16) :	10, 11, 12, 13, 14, 15	Bitiş değeri hiçbir zaman diziye dahil edilmez.
for i in range(0, 9, 2) :	0, 2, 4, 6, 8	Üçüncü değişken adım büyüklüğüdür.
for i in range(5, 0, -1) :	5, 4, 3, 2, 1	Geri yönde gitmek için negatif bir adım büyüklüğü kullanılır.

Yaygın Kullanılan For Döngüsü Algoritmaları

- Bir dizi girdinin toplamının hesaplanması

```
listem = [1, 6, 7, 9, 3, 4]
toplama = 0.0
for sayi in listem:
    toplama = toplama + sayi
print(toplama)
```

- Ortalamanın hesaplanması

```
listem = [1, 6, 7, 9, 3, 4]
toplama = 0.0
for sayi in listem:
    toplama += sayi
if len(listem) > 0:
    ortalama = toplama / len(listem)
else:
    ortalama = 0.0
print(ortalama)
```

Bazı Önemli İfadeler

14

- Pass: If komutlarının ardından boş ifade olamaz, ancak herhangi bir nedenle eylem içermeyen bir if komutu varsa, hata almaktan kaçınmak için pass kullanılır.

```
m = 100
n = 50
if n > m:
    print("n m'den büyüktür")
else:
    pass
```

- Break: Tüm elemanlar için döngü tamamlanmadan önce döngü durdurulabilir

```
listem = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in listem:
    print(i)
    if i == 5:
        break
```

- Continue: Döngünün belirli bir iterasyonunu atlayarak bir sonraki iterasyondan devam eder (pass gibi)

```
listem = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in listem:
    if i == 5:
        continue
    print(i)
```

MUDANYA
UNIVERSITY



Uygulama Soruları



Python'da Döngü Yapıları

Örnek Kullanıcı tarafından girilen iki sayı arasındaki çift sayıları yazdıran Python programını yazınız

```
ilk_deger = int(input("ilk degeri girin: "))
son_deger = int(input("son degeri girin: "))
for i in range(ilk_deger, son_deger+1):
    if i % 2 == 0:
        print(i)
```

Python'da Döngü Yapıları

17

Örnek Kullanıcı tarafından girilen iki sayı arasındaki tek sayıların toplamını yazdıran Python programını yazınız

```
ilk_deger = int(input("ilk degeri girin: "))
son_deger = int(input("son degeri girin: "))
toplam = 0
for i in range(ilk_deger, son_deger+1):
    if i % 2 == 1:
        toplam += i
print(toplam)
```


Python'da Döngü Yapıları

Örnek Kullanıcı tarafından girilen bir sayının faktöriyelini hesaplayan Python programını yazınız

Dikkat! Öncelikle girilen sayının negatif olmadığını kontrol ediniz.

```
kontrol = False
while not kontrol:
    sayi = int(input("sayiyi girin: "))
    if sayi < 0:
        print("gecersiz deger")
    else:
        kontrol = True
        faktoriyel = 1
        for i in range(1, sayi+1):
            faktoriyel *= i # faktoriyel = faktoriyel * i ile ayni anlamda
        print(f"{sayi} != {faktoriyel}")
```

Python'da Döngü Yapıları

19

Örnek 1'den kullanıcı tarafından girilen bir sayıya kadar olan sayıları ve her sayının karesini yazan Python programını yazınız (hem while hem de for yapısı kullanarak yapınız)

```
sayi = int(input("bir sayi giriniz: "))  
i = 1  
while i <= sayi:  
    kare = i ** 2  
    print(i, kare)  
    i += 1
```

```
sayi = int(input("bir sayi giriniz: "))  
for i in range(1, sayi+1):  
    kare = i ** 2  
    print(i, kare)
```

Python'da Döngü Yapıları

Örnek Kullanıcı negatif bir sayı girene kadar kullanıcının sayı girmesine izin veren ve ardından girilen tüm değerleri yazan programın Python kodunu yazın.

```
sayilar = []
while True:
    sayi = float(input("bir sayi girin (cikmak icin negatif sayi girin): "))
    if sayi < 0:
        break
    sayilar.append(sayi)
if len(sayilar) > 0:
    print("girilen sayilar:")
    for i in sayilar:
        print(i)
else:
    print("hic sayi girilmedi")
```