

PYTHON PROGRAMLAMA DİLİ

DÖNGÜLER
range-for-while-continue-break

Dr. Öğr. Üyesi M. Ozan AKI

DÖNGÜLER

Döngüler, bir kodu ya da kod bloğunu belir bir sayıda ya da belirli bir koşul sağlanana dek tekrar tekrar çalıştırılmasını sağlayan yapılardır.

Python dilinde iki tür döngü vardır;
Eleman sayısı belirli olan bir dizi üzerinde işlem yapmak için

for

Döngünün devam edip etmemesi sadece koşula bağlı olan

while

Eğer for döngüsünü List (Liste), Tuple (Demet), Set (Küme) ya da Dictionary (Sözlük) üzerinde çalıştırmayacaksak,

range

ile bir nesne oluşturup, belirlediğimiz düzende sayı dizilimine sahip bu nesne elemanları üzerinde döngümüzü çalıştırırız.

DÖNGÜLER - RANGE

Range, başlangıç ve bitiş noktası belirli bir sayı aralığı tanımlamak için kullanılır.

range(x)

0'dan **x**'e kadar sayılar.

`range(5) : 0 1 2 3 4`

`range(0,5) : 0 1 2 3 4`

`range(2,10) : 2 3 4 5 6 7 8 9`

`range(5,10,2) : 2 3 4 5 6 7 8 9`

`range(10,50,5) : 10 15 20 25 30 35 40 45`

`range(10,2,-1) : 10 9 8 7 6 5 4 3`

`range(20,5,-3) : 20 17 14 11 8`

`range(100,0,-10) : 100 90 80 70 60 50 40 30 20 10`

range(a,b)

a'dan **b**'ye kadar sayılar.

range(a,b,i)

a'dan **b**'ye kadar **i** artımlı sayılar.

DÖNGÜLER - FOR

Oluşturduğumuz **range** nesnesi in operatörü ile birlikte **for** döngüsünde kullanılır.

for i in range(a,b):
döngü kodları

range ile oluşturulan sayı dizisinin her bir elemanı sırayla döngü değişkeni olan **i**'ye atanır ve döngü kodları, **range** elemanları bitene kadar çalıştırılır.

```
>>> for i in range(5):
...     print(f"{i} ",end='')
...
0 1 2 3 4
>>>
>>> for i in range(5,10):
...     print(f"{i} ",end='')
...
...
5 6 7 8 9
>>> for i in range(5,30,5):
...     print(f"{i} ",end='')
...
...
5 10 15 20 25
>>> for i in range(12,2,-2):
...     print(f"{i} ",end='')
...
...
12 10 8 6 4
>>>
```

DÖNGÜLER - WHILE

while döngüsü, belirtilen bir koşul doğru olduğu sürece kod bloğundaki kodları tekrar tekrar çalıştırmaya devam eden bir yapıdır.

Döngü koşulu, döngü kodları içerisinde istenen sonuç elde edilince güncellenerek sonlandırılmalıdır.

Aksi halde döngüden hiç bir zaman çıkamaz ve program tepki veremez. Kodun Ctrl+C ile kesilerek sonlandırılması gerekir.

while *koşul:*

döngüde çalıştırılacak kodlar

döngüde çalıştırılacak kodlar

...

DÖNGÜLER - WHILE

while döngüsünde belirtilecek koşu, True ya da False değeri döndüren ifadeler olmalıdır.

Bu değerleri döndüren Karşılaştırma Operatörleri kullanılır.

```
>>> i=0
...
>>> while i<5:
...     print(f"{i} ",end='')
...     i+=1
...
0 1 2 3 4
>>>
>>> i=20
>>> while i>5:
...     print(f"{i} ",end='')
...     i-=2
...
20 18 16 14 12 10 8 6
>>>
```

DÖNGÜLER - BREAK

for ya da **while** döngüsü çalışmaya devam ederken, döngü kodunun herhangi yerinde döngü koşulunu beklemeden bir anda döngüyü sonlandırıp döngüden çıkmak için;

break

İfadesi kullanılır.

Genelde **break** ifadesi bir **if** koşulu altında kullanılır.

```
>>> for i in range(100):
...     print(f"{i} ",end='')
...     if i==9:
...         break
...
0 1 2 3 4 5 6 7 8 9
>>>
>>> i=0
>>> while i<100:
...     print(f"{i} ",end='')
...     if i==9:
...         break
...     i+=1
...
0 1 2 3 4 5 6 7 8 9
>>> |
```

DÖNGÜLER - CONTINUE

for ya da **while** döngüsü çalışmaya devam ederken, koşulun tekrar değerlendirilmesi gerektiği bir anda kod akışını döngünün başına göndermek için;

continue

İfadesi kullanılır.

Genelde **continue** ifadesi bir **if** koşulu altında kullanılır.

```
>>> for i in range(20):
...     if i%2==0:
...         continue
...     print(f"{i} ",end='')
...
...
1 3 5 7 9 11 13 15 17 19
>>>
>>> i=0
>>> while i<20:
...     i+=1
...     if i%3==0:
...         continue
...     print(f"{i} ",end='')
...
...
1 2 4 5 7 8 10 11 13 14 16 17 19 20
>>> |
```


DÖNGÜLER – SONSUZ DÖNGÜ

Bir mantık hatası sonucu hiç bir zaman sonlanmayan döngü oluştuğunda program akışı döngü içerisine hapsolür ve yazdığımız uygulama cevap veremez.

Kullanıcının gözünde program kilitlenmiştir. Ya görev yöneticisinden görevi sonlandırılır ya da bilgisayar yeniden başlatılır.

Bu gibi durumlarda kaçınmak için özellikle while döngüsünü kullanırken dikkatli olmalıyız.

```
i=0
while i<100:
    print(f"{i} ", endl='')
```

Bu bir sonsuz döngüdür. Çünkü i değişkenini döngü içerisinde arttırma kodu unutulmuştur. Yorumlayıcı hata vermez çünkü yazımsal olarak bir sorun yoktur. Buradaki hata mantık (algoritma) hatasıdır.