

Python Programlama Dili – Döngüler

En son makalelerimizde Python bileşenleri için temellerini attık ve bu yazımızda da döngüleri ele alacağız.

Döngüler

Döngüler programınızda birden fazla tekrar eden durumlarda, belli şartları sağladığı takdirde işlev gören durumlarda kullandığımız ifadelerdir... Bu ifadeler bizim sürekli işlemleri tekrarlamamızdan ziyade işimizi tek satır kodda halletmemizi sağlayan yapılardır.

Örneğin bir saat içinde her dakikada ekrana mesaj yazdırmak istediğinizde mantık;

```
bir dakika bekle
  ekrana mesaj yaz
bir dakika bekle
  ekrana mesaj yaz
...
bir dakika bekle
  ekrana mesaj yaz
```

şeklinde 60 kere tek tek yazmanız gerekecek, fakat bunun yerine;

```
while 60dk bitti
  ekrana mesaj yaz
bir dakika bekle
```

şeklinde bir döngüyle kısaltabiliriz. 1den 100'e kadar tüm sayıları ekrana yazdırmak için

```
print 1
print 2
print 3
...
print 99
print 100
```

şeklinde tek tek yazmak gerekirdi. Bunun yerine döngülerle işimiz çok daha kolaylaşır.

While Döngüsü

```
while <koşul>:
  <durum>
```

while döngüsünün temel prensibi “koşul”un sağlandığı takdirde “durum” un gerçekleşmesidir.

```
x=1
while x<100:
  print x
  x=x+1
```

```
>>> isim=""
>>> while not isim:
    isim=raw_input('Lütfen Adınızı giriniz...')
    print 'Merhaba,%s!' % isim
```



Lütfen Adınızı giriniz...ibrahim
Merhaba,ibrahim!

For Döngüsü

```
>>> kelimeler=['Bu','Bir','Deneme','Döngüsü']  
>>> for kelime in kelimeler:  
    print kelime
```

Bu
Bir
Deneme
Döngüsü

range()

range() komutu bizim için otomatik dizi oluşturmaya yarayan bir komuttur. Bizim için parametreyi alır ve istediğimiz bir koşulda bir liste oluşturur.

range(bitiş)

formunda kullanırsak range komutunu 0'dan başlayarak bitiş sayısına kadar birer artan bir dizi oluşturur.

```
>>> range(29)  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
```

döngü içerisinde kullanmak bize çok kolaylıklar sağlar;

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

0
1
2
3
4

range(başlangıç,bitiş,artış)

range komutunun bir değişik kullanımı da yukarıdaki formdadır. Başlangıç bitiş ve artış parametrelerini girdikten sonra otomatikman listeyi ona göre oluşturur.

```
>>> range(1,10,2)  
[1, 3, 5, 7, 9]
```

range(başlangıç,bitiş)

bu from da sayılar başlangıçtan bitişe kadar tek tek artar.

```
>>> range(10,14)  
[10, 11, 12, 13]
```

Break

Döngülerden çıkmak için kullanılır.

```
From math import sqrt  
for n in range(99,0,-1):  
    root=sqrt(n)  
    if root == int(root)  
        print n  
        break
```



Continue

döngülerde devam etmeyi sağlar.

If Statement

```
if <koşul>:  
    <durum>  
elif <koşul>:  
    <durum>  
elif <koşul>:  
    <durum>  
...  
else:  
    <durum>
```

Örnek olarak, önceden belirlenmiş bir sayıyı tahmin etme döngüsü yazalım;

```
if girilen_sayı <100:  
    print 'Daha büyük bir sayı giriniz...'  
elif girilen_sayı >100:  
    print 'Daha küçük bir sayı giriniz...'  
elif girilen_sayı ==100:  
    print 'Tahmininiz doğru! 100!'  
else:  
    print 'Yanlış karakter! Lütfen sayı giriniz.'
```

Hepinize Kolay gelsin. Sorularınızı ve takıldığınız yerleri ik@ibrahimkivanc.com mail adresime iletebilirsiniz.

