

DÖNGÜLER(LOOPS)

Döngü yapısı

For döngüsü ve kullanımı

FOR Döngüsünün kullanılması

- For döngüleri belirlirtilen sayıdaki işlemlerin tekrarlanması için kullanılan döngülerdir. For döngüleri başlangıç ve bitiş değerleri arasında artım miktarına göre istenilen sayıda tekrar yapar.

For değişken **in** değer-aralık:

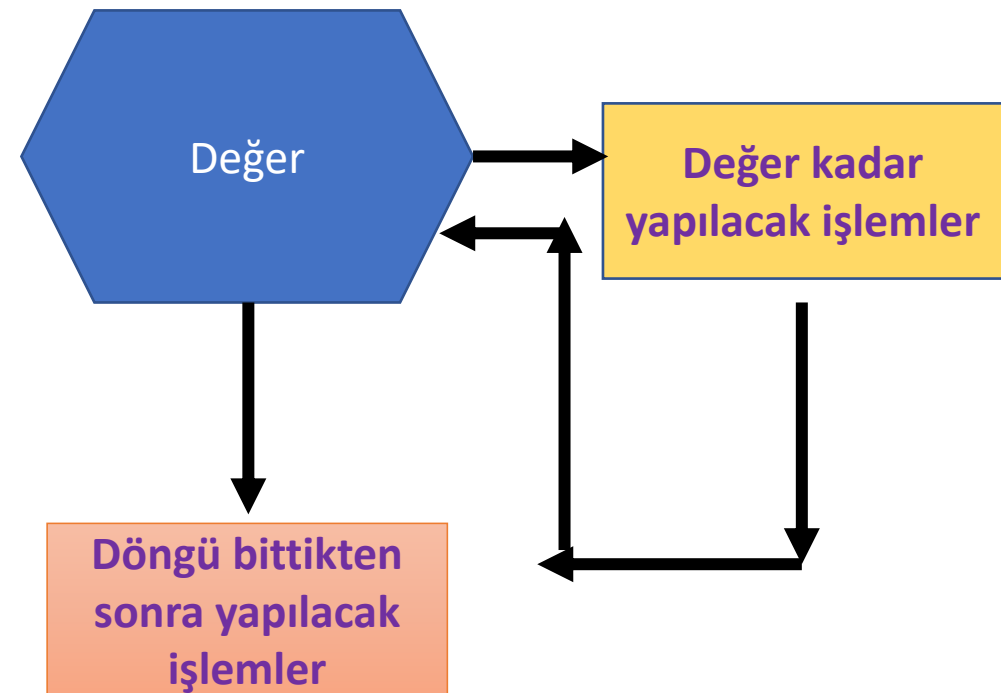
Yapılacak işlem

Yapılacak işlem

Yapılacak işlem

.

.



range () fonksiyonunun kullanılması

Range aralık demektir. Range fonksiyonunu kullanarak belirtilen aralıktaki sayıları for döngüsünde kullanabiliriz.

Range fonksiyonunun üç kullanımı vardır

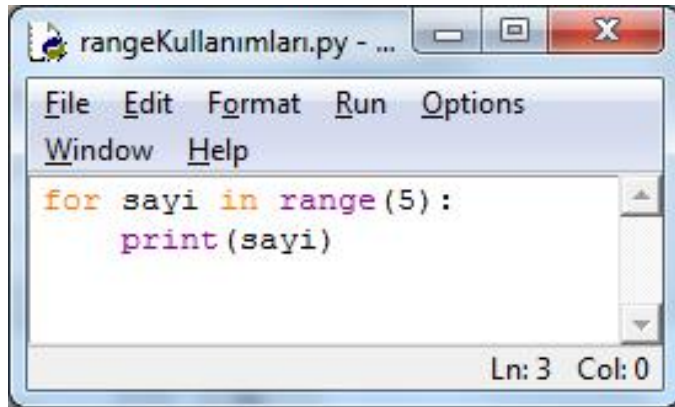
`range(değer)`

`range(başlangıç, bitiş)`

`range(başlangıç, bitiş, artış değeri)`

range(değer) kullanımı

0 dan başlar, değerın 1 eksiğine kadar sayı üretir.
Döngünün tur sayısı ise belirtilen sayı kadardır.

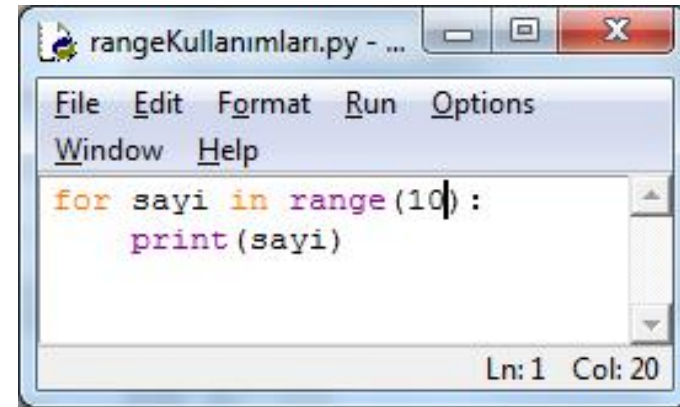


```
File Edit Format Run Options
Window Help

for sayi in range(5):
    print(sayi)

Ln: 3 Col: 0
```

```
0
1
2
3
4
>>>
```



```
File Edit Format Run Options
Window Help

for sayi in range(10):
    print(sayi)

Ln: 1 Col: 20
```

```
0
1
2
3
4
5
6
7
8
9
>>>
```

range kullanımında bitiş değeri döngüye dahil edilmez

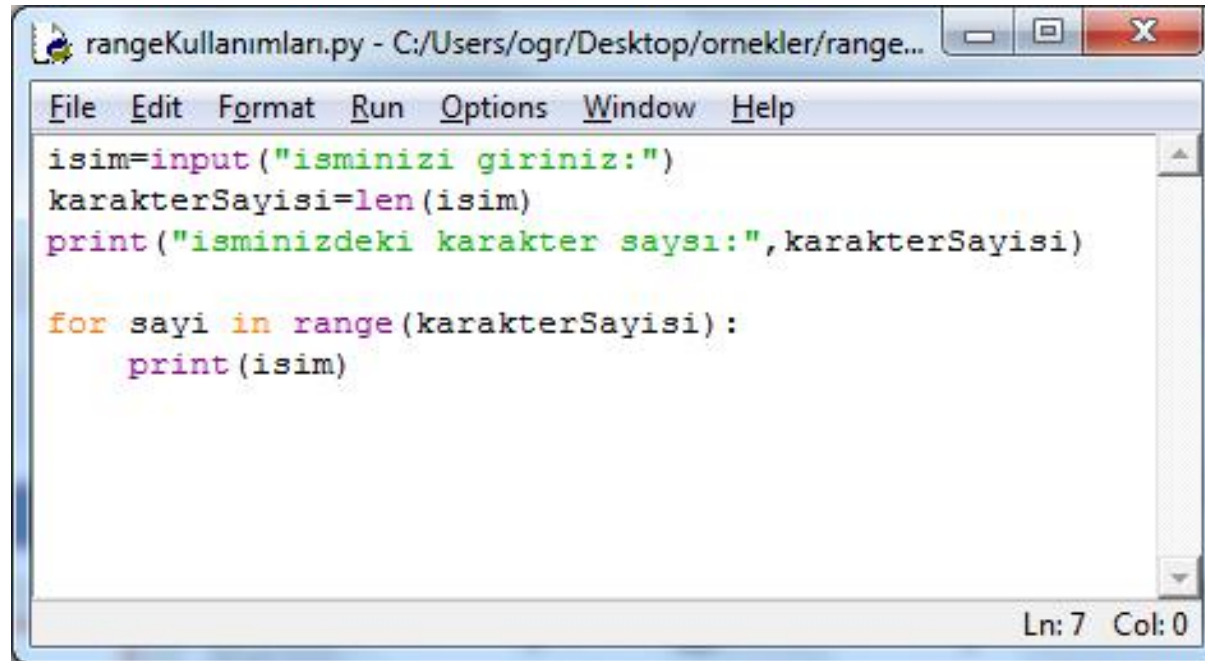
range(5)= 0,1,2,3,4,

range(6)=0,1,2,3,4,5

range(7)=0,1,2,3,4,5,6

Değerlerini alır

Girilen isimdeki harf kadar ,ismi yazan programı yazalım



```
rangeKullanımları.py - C:/Users/ogr/Desktop/ornekler/range...  
File Edit Format Run Options Window Help  
isim=input("isminizi giriniz:")  
karakterSayisi=len(isim)  
print("isminizdeki karakter sayısı:",karakterSayisi)  
  
for sayi in range(karakterSayisi):  
    print(isim)  
  
Ln: 7 Col: 0
```

```
isminizi giriniz:anibal  
isminizdeki karakter sayısı: 6  
anibal  
anibal  
anibal  
anibal  
anibal  
anibal  
anibal  
>>> |
```

`range(başlangıç, bitiş)` Kullanımı
başlangıç değeri ile başlar, bitiş değerinin 1 eksiğine
kadar döngü devam eder.

range
kullanımında
bitiş değeri
döngüye dahil
edilmez

```
forDöngüsü.py - C:...
File Edit Format Run Options
Window Help
for i in range(1,11):
    print(i)
Ln: 1 Col: 14
```

```
rangeKullanımları.py - ...
File Edit Format Run Options
Window Help
for sayi in range(3,10):
    print(sayi)
Ln: 1 Col: 20
```

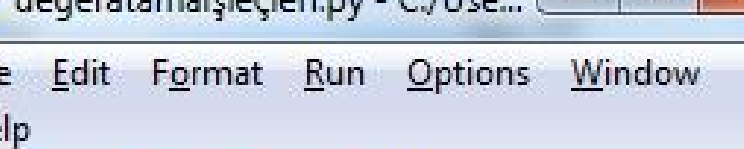
```
*rangeKullanımları.py - ...
File Edit Format Run Options
Window Help
for sayi in range(7,15):
    print(sayi)
Ln: 1 Col: 23
```

```
1
2
3
4
5
6
7
8
9
10
>>> |
```

```
3
4
5
6
7
8
9
>>> |
```

```
7
8
9
10
11
12
13
14
>>> |
```

Şimdi «anibal» yazısını for döngüsünü kullanarak 100 defa yazdıralım



*değeratama işleçleri.py - C:/Use...

File Edit Format Run Options Window Help

```
for i in range(1,101):  
    print("anibal")
```

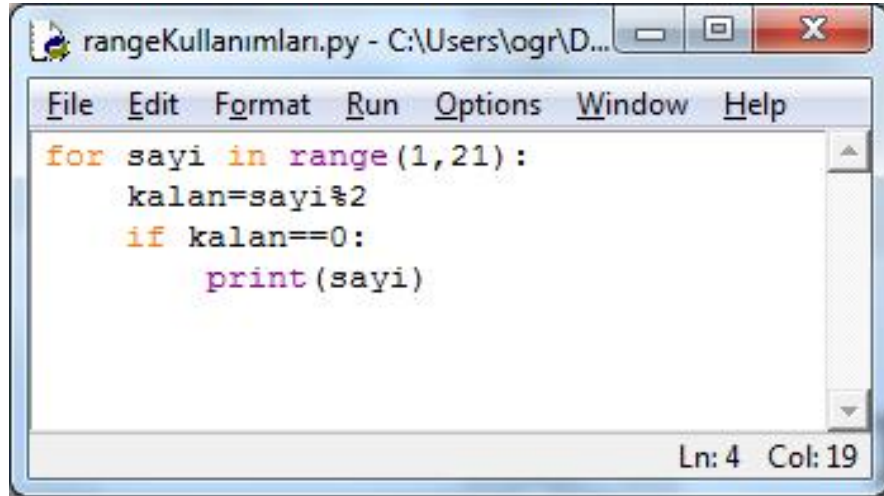
Ln: 2 Col: 10

Bu işlem 100 defa «anibal» yazar

[illegible]

range fonksiyonunda bitiş değeri döngüye Dahil değildir

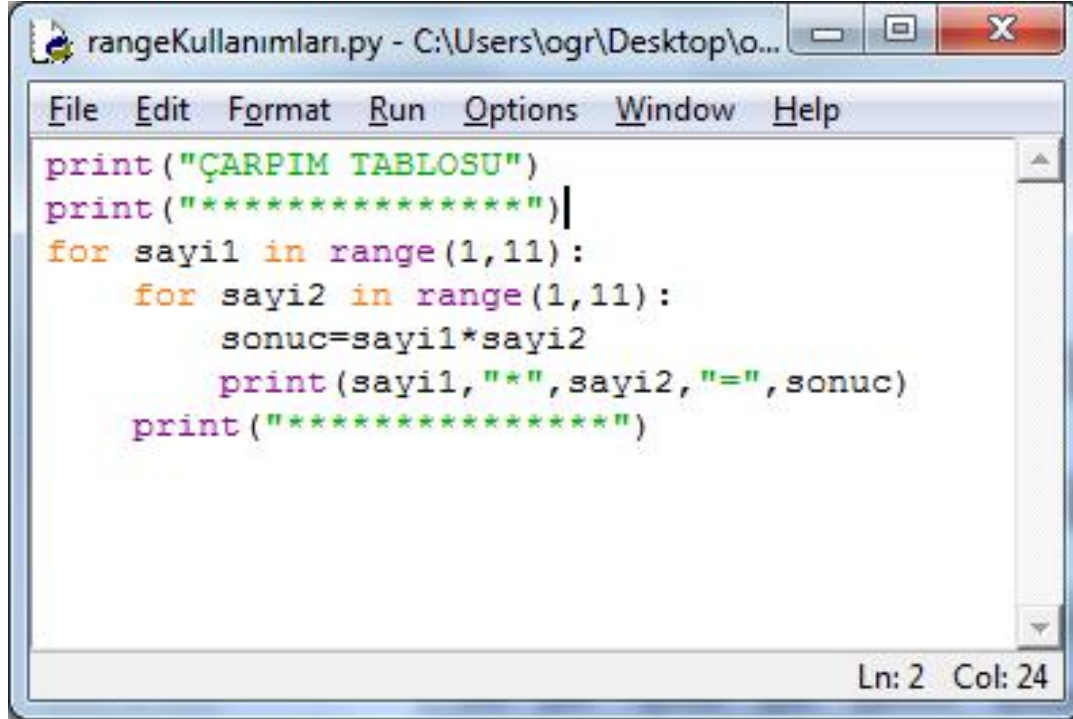
1 den 20' ye kadar(20 dahil) olan çift sayıları yazdıralım



```
rangeKullanımları.py - C:\Users\ogr\D...  
File Edit Format Run Options Window Help  
for sayi in range(1,21):  
    kalan=sayi%2  
    if kalan==0:  
        print(sayi)  
Ln: 4 Col: 19
```

```
2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
>>> |
```


ÇARPIM TABLOSU YAPALIM(İç içe for döngüsü kullanımı)

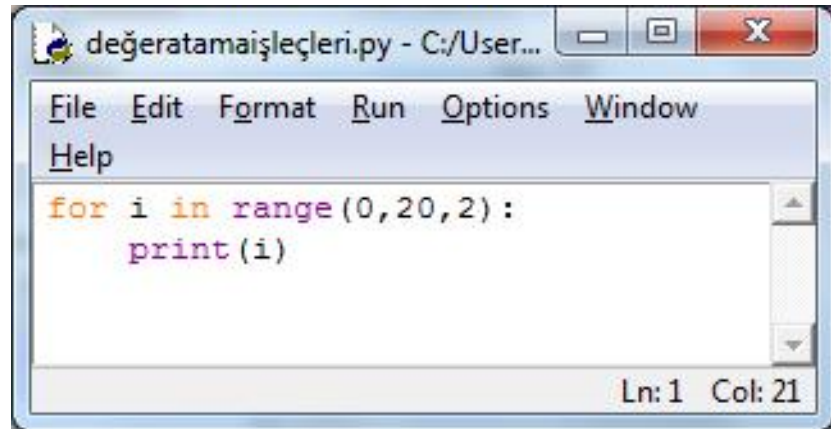


```
rangeKullanımları.py - C:\Users\ogr\Desktop\o...
File Edit Format Run Options Window Help
print("ÇARPIM TABLOSU")
print("*****")
for sayi1 in range(1,11):
    for sayi2 in range(1,11):
        sonuc=sayi1*sayi2
        print(sayi1,"*",sayi2,"=",sonuc)
    print("*****")
Ln: 2 Col: 24
```

ÇARPIM TABLOSU

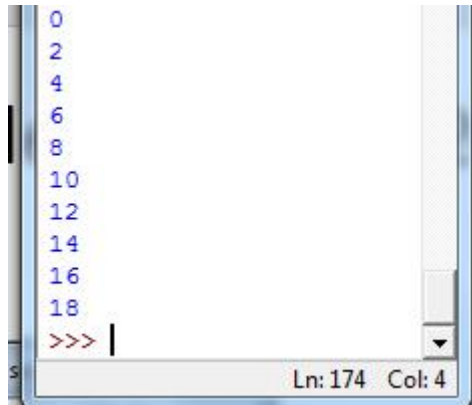
```
*****
1 * 1 = 1  4 * 1 = 4  7 * 1 = 7  10 * 1 = 10
1 * 2 = 2  4 * 2 = 8  7 * 2 = 14 10 * 2 = 20
1 * 3 = 3  4 * 3 = 12 7 * 3 = 21 10 * 3 = 30
1 * 4 = 4  4 * 4 = 16 7 * 4 = 28 10 * 4 = 40
1 * 5 = 5  4 * 5 = 20 7 * 5 = 35 10 * 5 = 50
1 * 6 = 6  4 * 6 = 24 7 * 6 = 42 10 * 6 = 60
1 * 7 = 7  4 * 7 = 28 7 * 7 = 49 10 * 7 = 70
1 * 8 = 8  4 * 8 = 32 7 * 8 = 56 10 * 8 = 80
1 * 9 = 9  4 * 9 = 36 7 * 9 = 63 10 * 9 = 90
1 * 10 = 10 4 * 10 = 40 7 * 10 = 70 10 * 10 = 100
*****
2 * 1 = 2  5 * 1 = 5  8 * 1 = 8
2 * 2 = 4  5 * 2 = 10 8 * 2 = 16
2 * 3 = 6  5 * 3 = 15 8 * 3 = 24
2 * 4 = 8  5 * 4 = 20 8 * 4 = 32
2 * 5 = 10 5 * 5 = 25 8 * 5 = 40
2 * 6 = 12 5 * 6 = 30 8 * 6 = 48
2 * 7 = 14 5 * 7 = 35 8 * 7 = 56
2 * 8 = 16 5 * 8 = 40 8 * 8 = 64
2 * 9 = 18 5 * 9 = 45 8 * 9 = 72
2 * 10 = 20 5 * 10 = 50 8 * 10 = 80
*****
3 * 1 = 3  6 * 1 = 6  9 * 1 = 9
3 * 2 = 6  6 * 2 = 12 9 * 2 = 18
3 * 3 = 9  6 * 3 = 18 9 * 3 = 27
3 * 4 = 12 6 * 4 = 24 9 * 4 = 36
3 * 5 = 15 6 * 5 = 30 9 * 5 = 45
3 * 6 = 18 6 * 6 = 36 9 * 6 = 54
3 * 7 = 21 6 * 7 = 42 9 * 7 = 63
3 * 8 = 24 6 * 8 = 48 9 * 8 = 72
3 * 9 = 27 6 * 9 = 54 9 * 9 = 81
3 * 10 = 30 6 * 10 = 60 9 * 10 = 90
*****
```

`range(başlangıç,bitiş,artış değeri)` kullanımı
başlangıç değerinden ,bitiş değerinin 1 eksiğine
kadar,artış değerince döngü devam eder



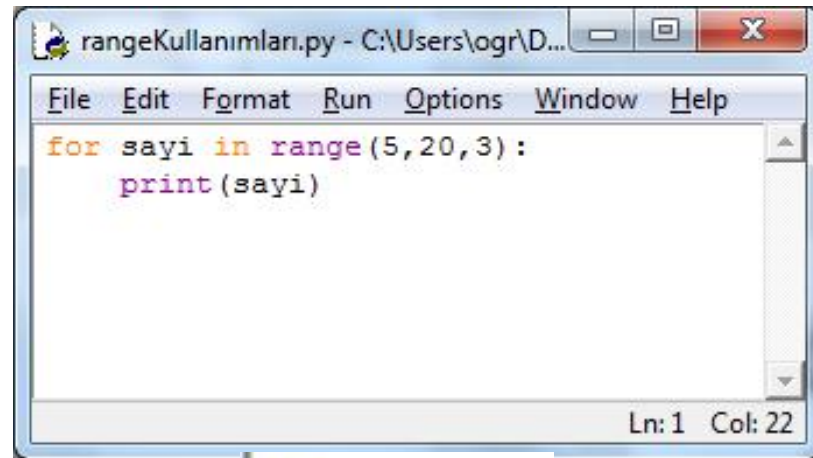
```
File Edit Format Run Options Window Help
for i in range(0,20,2):
    print(i)
```

Ln: 1 Col: 21



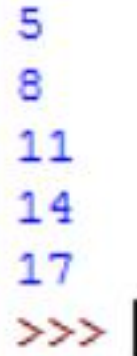
```
0
2
4
6
8
10
12
14
16
18
>>> |
```

Ln: 174 Col: 4



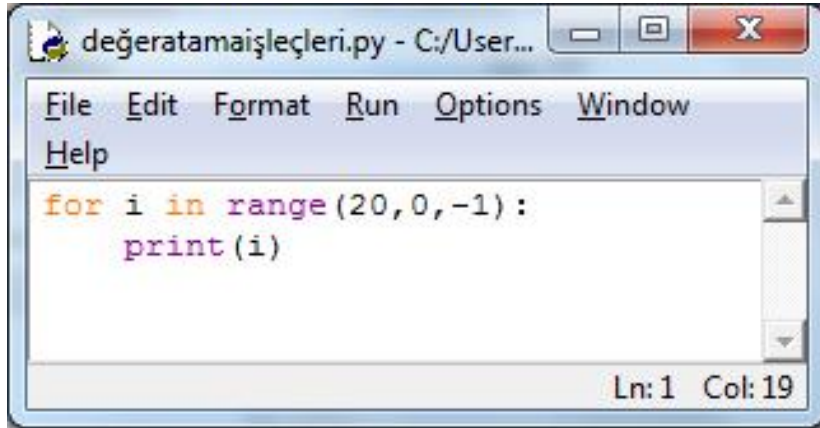
```
File Edit Format Run Options Window Help
for sayi in range(5,20,3):
    print(sayi)
```

Ln: 1 Col: 22



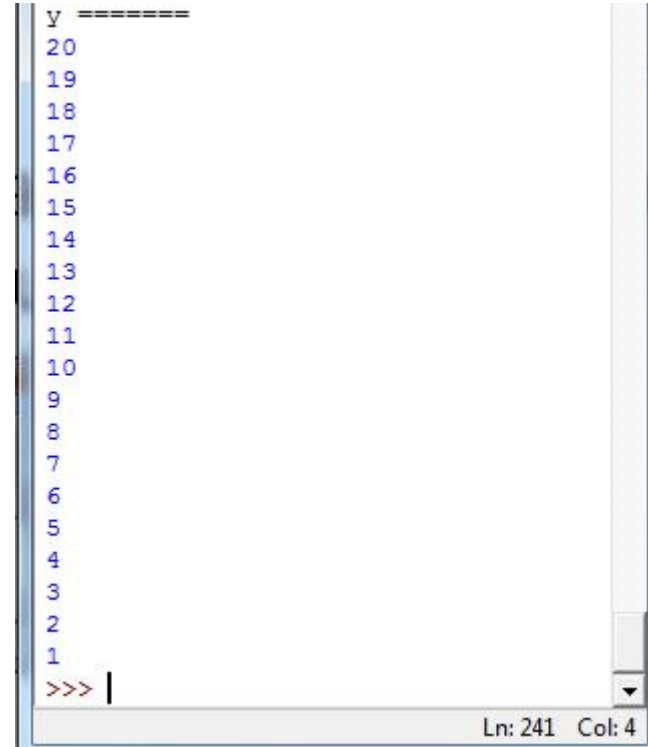
```
5
8
11
14
17
>>> |
```

Örnek: 20 den 0'a kadar 1'er geri sayan for döngüsünü oluşturalım



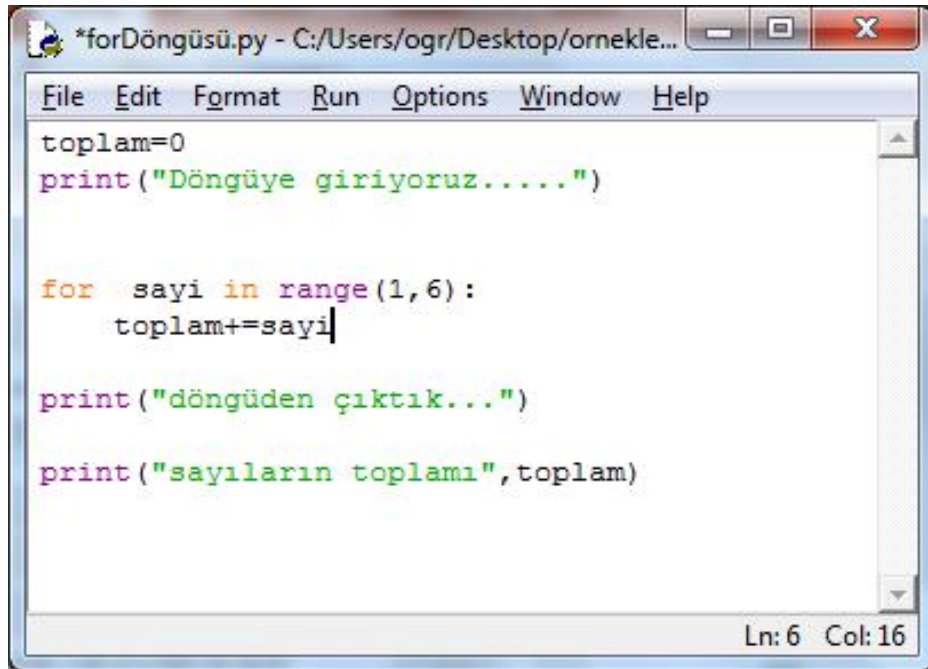
```
değeratamaışleçleri.py - C:/User...  
File Edit Format Run Options Window  
Help  
for i in range(20,0,-1):  
    print(i)  
Ln:1 Col:19
```

`range(başlangıç,bitiş,artış değeri)`



```
Y =====  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
>>> |  
Ln: 241 Col: 4
```

Örnek: 1 den 5 'a kadar olan sayıların toplamını bulan programı yazalım



```
*forDöngüsü.py - C:/Users/ogr/Desktop/ornekle...
File Edit Format Run Options Window Help
toplam=0
print("Döngüye giriyoruz.....")

for sayi in range(1,6):
    toplam+=sayi

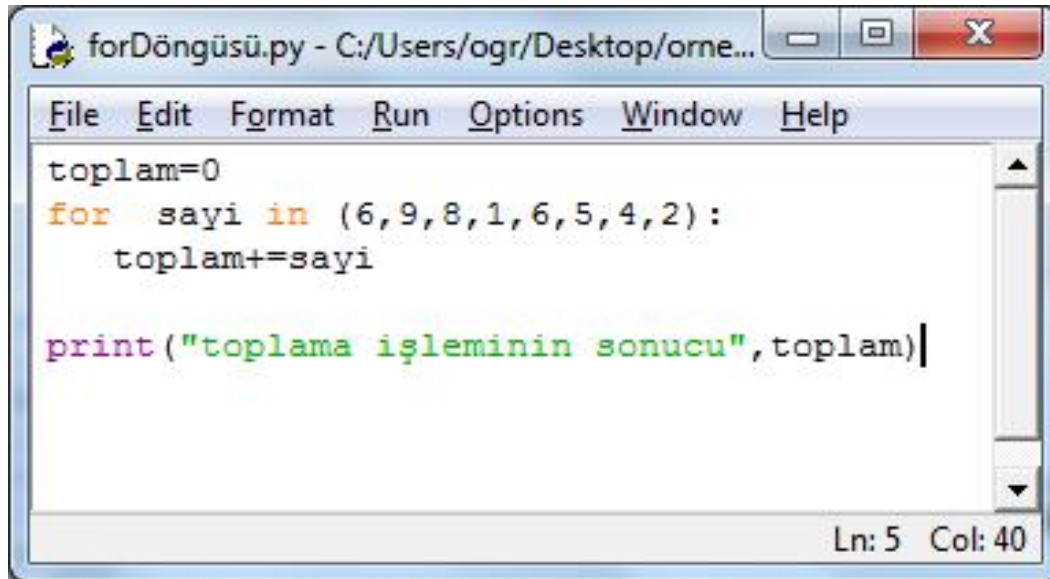
print("döngüden çıktık...")

print("sayıların toplamı",toplam)

Ln: 6 Col: 16
```

```
Döngüye giriyoruz.....
döngüden çıktık...
sayıların toplamı 15
>>> |
```

Sıralı olmayan sayıları toplayalım

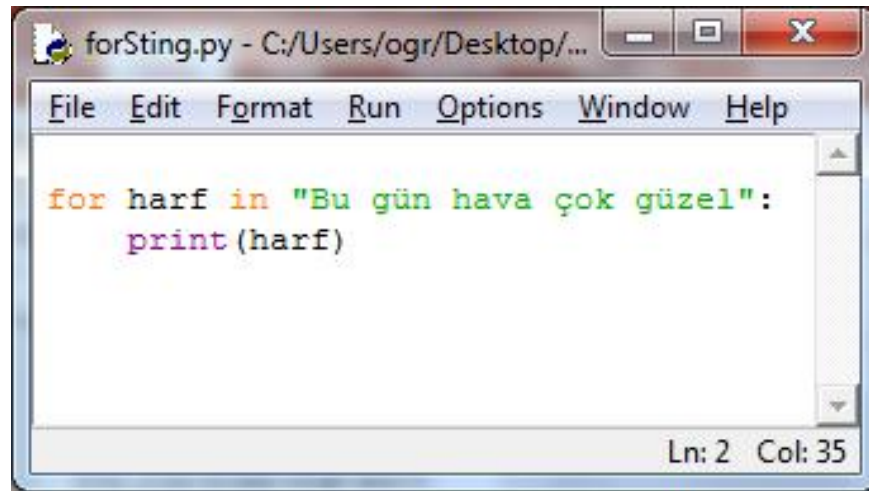


```
forDöngüsü.py - C:/Users/ogr/Desktop/orne...  
File Edit Format Run Options Window Help  
toplama=0  
for sayi in (6,9,8,1,6,5,4,2):  
    toplama+=sayi  
  
print("toplama işleminin sonucu",toplama)|  
Ln: 5 Col: 40
```

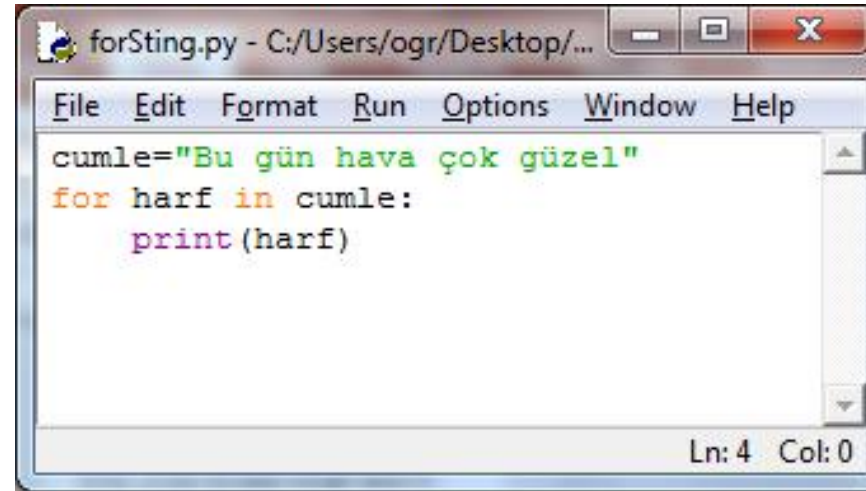
```
===== RESTART: C:/Users/  
toplama işleminin sonucu 41  
>>> |
```


For döngüsünde string verilerin kullanılması

For döngüsü ile string veri içerisindeki her bir karakteri alabiliriz



```
forSting.py - C:/Users/ogr/Desktop/...  
File Edit Format Run Options Window Help  
  
for harf in "Bu gün hava çok güzel":  
    print(harf)  
  
Ln: 2 Col: 35
```



```
forSting.py - C:/Users/ogr/Desktop/...  
File Edit Format Run Options Window Help  
  
cumle="Bu gün hava çok güzel"  
for harf in cumle:  
    print(harf)  
  
Ln: 4 Col: 0
```

B
u

g
ü
n

h
a
v
a

Ç
o
k

g
ü
z
e
l

>>>

Girilen cümle içerisindeki sesli harflerin sayısını bulalım

Algoritma:

1.Adım:Başla

2.adım:Sesli harfleri tanımla-ata

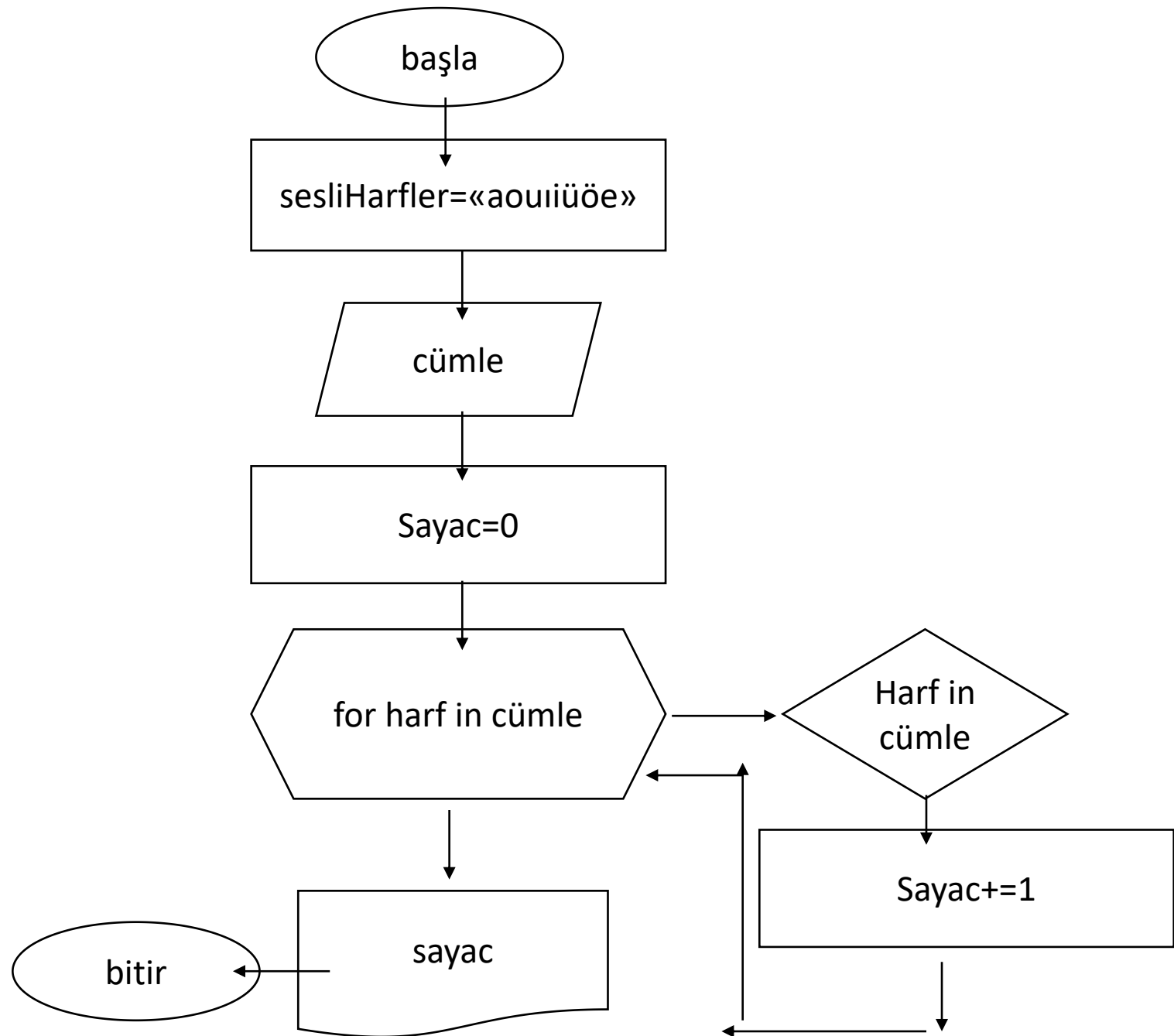
3.adım: bir cümle gir

4.Adım:Sesli harf sayısını tutması için bir sayaç tanımla

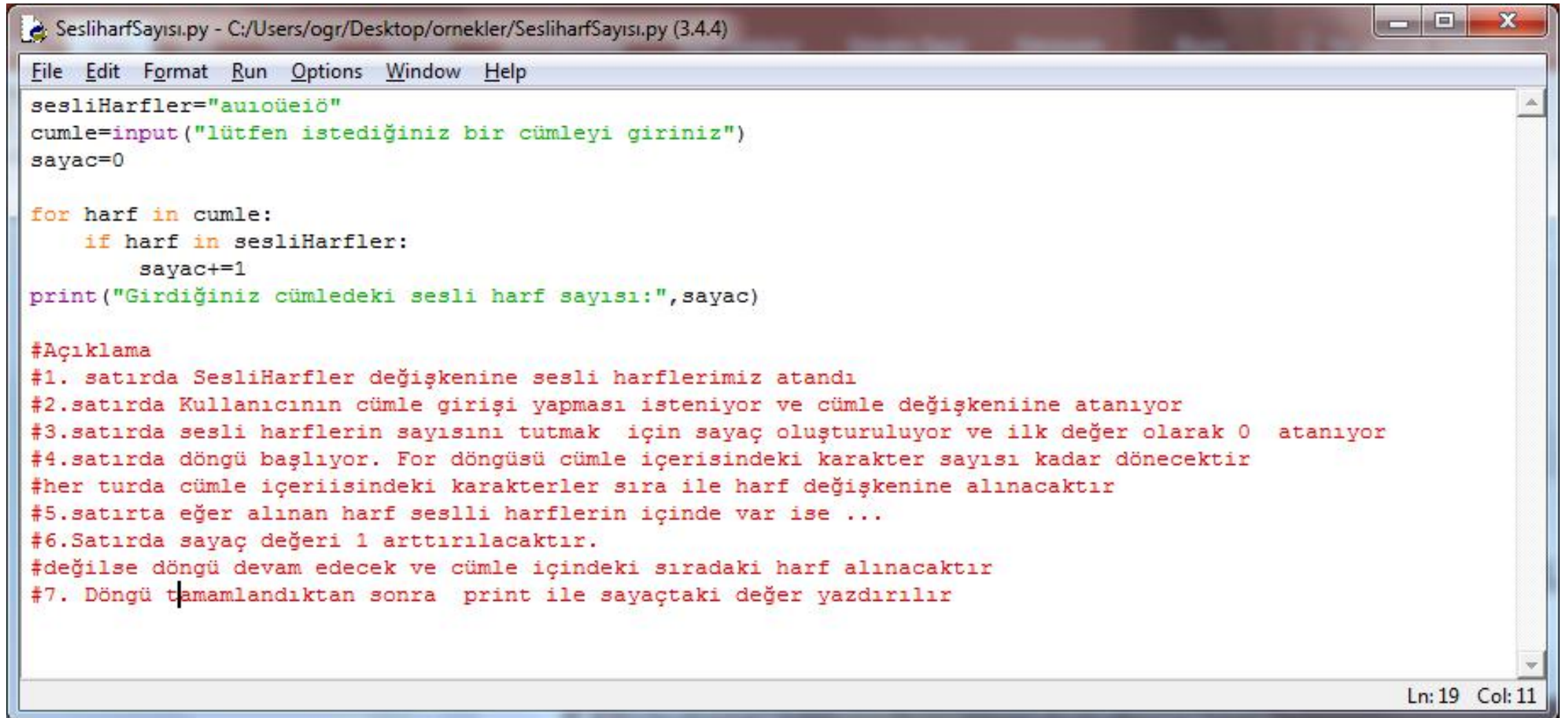
5.Adım :Cümleyi döngüye alarak harfleri sıra ile al

6.Adım:Eğer harfi sesli harflerin içinde var ise sayacı 1 arttır

7.Adım: Döngü Bittikten sonra sesli harf sayısını yazdır.



Girilen cümle içerisindeki sesli harflerin sayısını bulalım



```
SesliHarfSayisi.py - C:/Users/ogr/Desktop/ornekler/SesliHarfSayisi.py (3.4.4)
File Edit Format Run Options Window Help
sesliHarfler="auioüeiö"
cumle=input("lütfen istediğiniz bir cümleyi giriniz")
sayac=0

for harf in cumle:
    if harf in sesliHarfler:
        sayac+=1
print("Girdiğiniz cümledeki sesli harf sayısı:",sayac)

#Açıklama
#1. satırda SesliHarfler değişkenine sesli harflerimiz atandı
#2.satırda Kullanıcının cümle girişi yapması isteniyor ve cümle değişkenine atanıyor
#3.satırda sesli harflerin sayısını tutmak için sayaç oluşturuluyor ve ilk değer olarak 0 atanıyor
#4.satırda döngü başlıyor. For döngüsü cümle içerisindeki karakter sayısı kadar dönecektir
#her turda cümle içerisindeki karakterler sıra ile harf değişkenine alınacaktır
#5.satırda eğer alınan harf sesli harflerin içinde var ise ...
#6.Satırda sayaç değeri 1 arttırılacaktır.
#değilse döngü devam edecek ve cümle içindeki sıradaki harf alınacaktır
#7. Döngü tamamlandıktan sonra print ile sayaçtaki değer yazdırılır
```

Ln: 19 Col: 11

Döngüyü **break** komutu ile bitirme-kesme

Döngü devam ederken(tamamlanmadan) belli bir kriter gere göre döngüyü kesebilir ve bitirebiliriz.

ÖRNEK: 1 den 10 a kadar devam eden döngüyü .
5.Sayıya gelince kesen programı oluşturalım

NORMAL DÖNGÜ

```
brekfor.py - C:/Users/ogr/Desktop/orne...  
File Edit Format Run Options Window Help  
for sayi in range(1,11):  
    print(sayi)  
|
```

Ln: 3 Col: 0

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

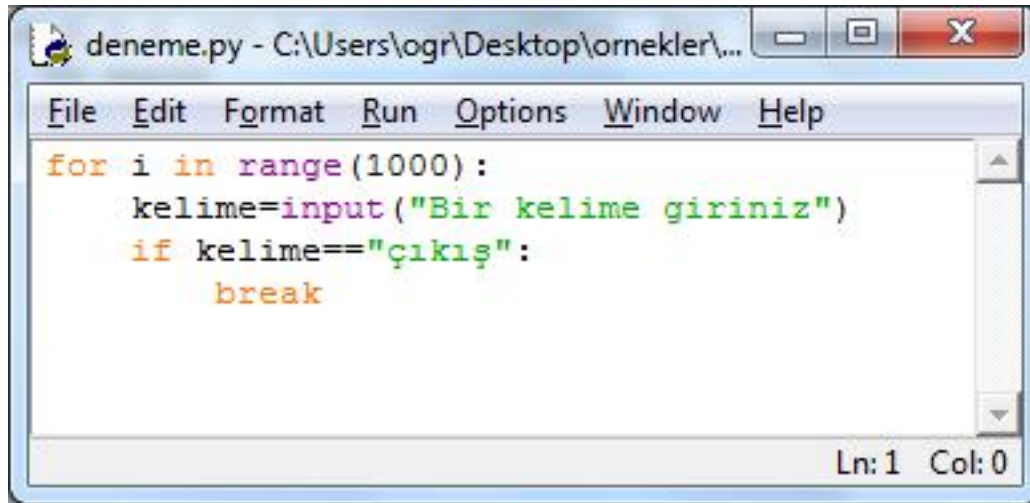
KIRILMIŞ DÖNGÜ

```
*brekfor.py - C:/Users/ogr/Desktop/orn...  
File Edit Format Run Options Window Help  
for sayi in range(1,11):  
    print(sayi)  
    if sayi==5:  
        break  
|
```

Ln: 3 Col: 15

```
1  
2  
3  
4  
5  
>>> |
```

Örnek: 1000 defa dönen döngü içerisinde girilen kelime «çıkış» ise döngüyü sonlandıran kodları yazalım

A screenshot of a Python IDE window titled 'deneme.py - C:\Users\ogr\Desktop\ornekler\...'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains the following Python code:

```
for i in range(1000):  
    kelime=input("Bir kelime giriniz")  
    if kelime=="çıkış":  
        break
```

The status bar at the bottom right shows 'Ln: 1 Col: 0'.

Yandaki örnek ile 1000 defa kelime girişi yapılabilir. Girilen kelime «çıkış» ise döngü ye son verilecektir.

Continue ile döngüyü başa sarma

Döngüler **continue** komutu ile karşılaştıklarında **continue** altındaki komutları çalıştırmadan döngüyü baştan devam ettirir

Örnek: 1 den 10 kadar (10 dahil) sayıları yazan döngüde 7 sayısını yazmayan program

```
for sayi in range(1,11):  
    if(sayi==7):  
        continue  
    print(sayi)
```

```
1  
2  
3  
4  
5  
6  
8  
9  
10  
>>> |
```

ÖRNEK: 50 nin altındaki sayıları continue ile yazmadan geçen program

```
for sayi in (50, 45, 12, 65, 89, 87, 45, 26, 55) :  
    if (sayi < 50) :  
        continue  
    print (sayi)
```

```
50  
65  
89  
87  
55  
>>> |
```