
Python ile Programlamaya Giriş

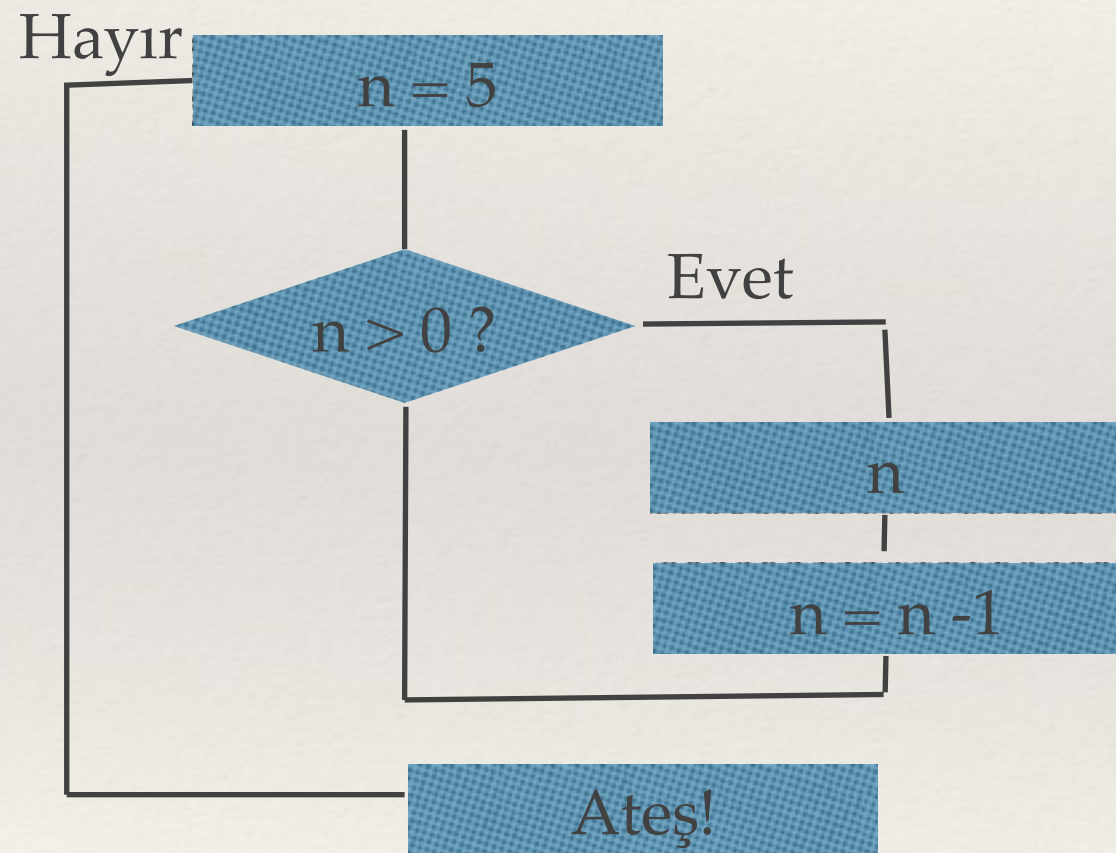
DERS 5: DÖNGÜLER

DR. HÜSEYİN BAHTİYAR



Döngüler ve İterasyon

- ❖ Döngülerin amacı, verilen koşulun gerçekleşmesi süresince belirli bir kodun tekrarlanmasını sağlamaktır.



Çıktısı:

5
4
3
2
1

Ateş!

Belirli / Belirsiz Döngüler

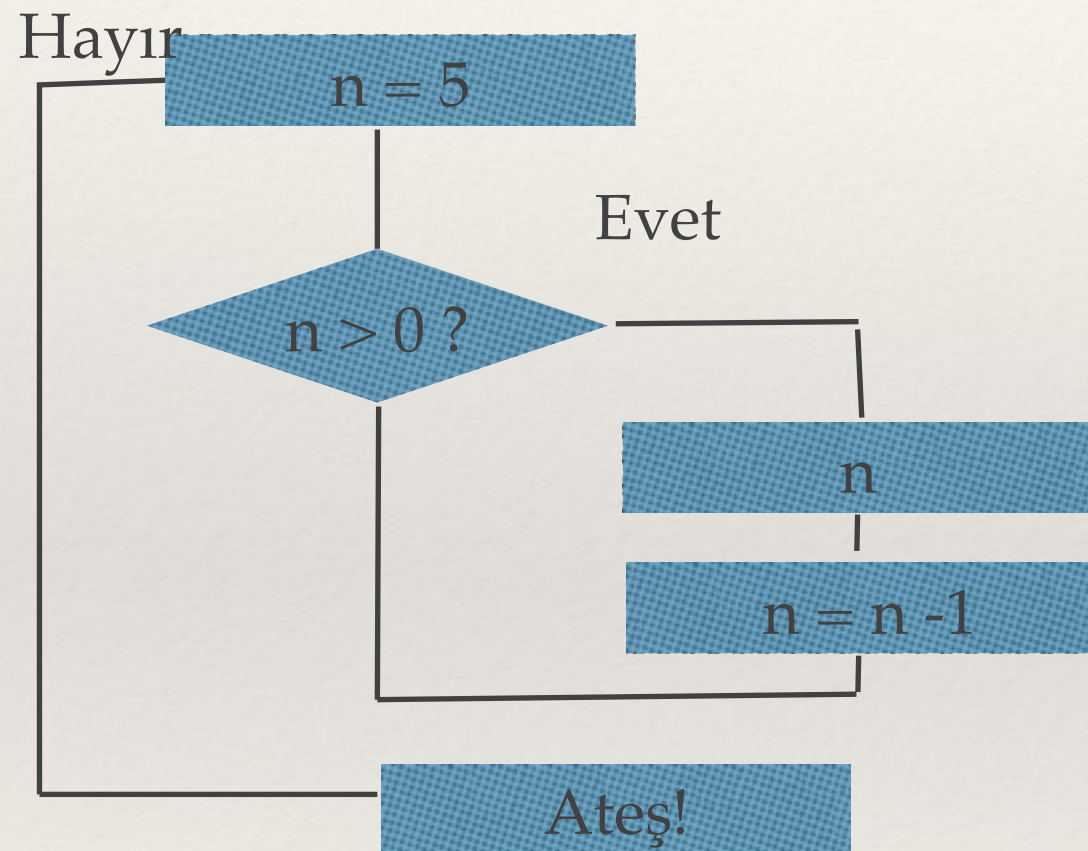
- ❖ Döngü yapıları ikiye ayrılır, belirli ve belirsiz döngüler.
 - ❖ Belirsiz döngülerde mantıksal bir koşul vardır ve o koşul **Yanlış** olana kadar döngü devam eder.
 - ❖ Eğer döngü sayısını bilmiyorsak bir koşul sonucunda döngüden çıkmasını istiyorsak **belirsiz döngüler** kullanmamız gerekir.
 - ❖ Örnek: Kullanıcı negatif sayı girene kadar sayı girmesini istemek.
 - ❖ Belirli Döngülerde ise döngü sayısını (sonlu sayıda) bildiğimiz yapılarda kullanırız.
 - ❖ Örnek: kullanıcıdan 5 kere üst üste sayı girmesini istiyorsak **belirli döngüler** kullanırız.

Belirsiz Döngüler

- ❖ Belirsiz döngüler için `while` komutunu kullanırız.
- ❖ While döngüsünde **önce koşul kontrol edilir** **doğruysa döngü işlemi yapılır.**
- ❖ `while` **koşul** :
 - ❖ Yapılmasını istediğimiz komutlar

```
#!/python3
n = input("sayi gir:")
while n>0:
    print("sayi pozitif")
    n=n-1|
```

Belirsiz Döngüler Örnekler:



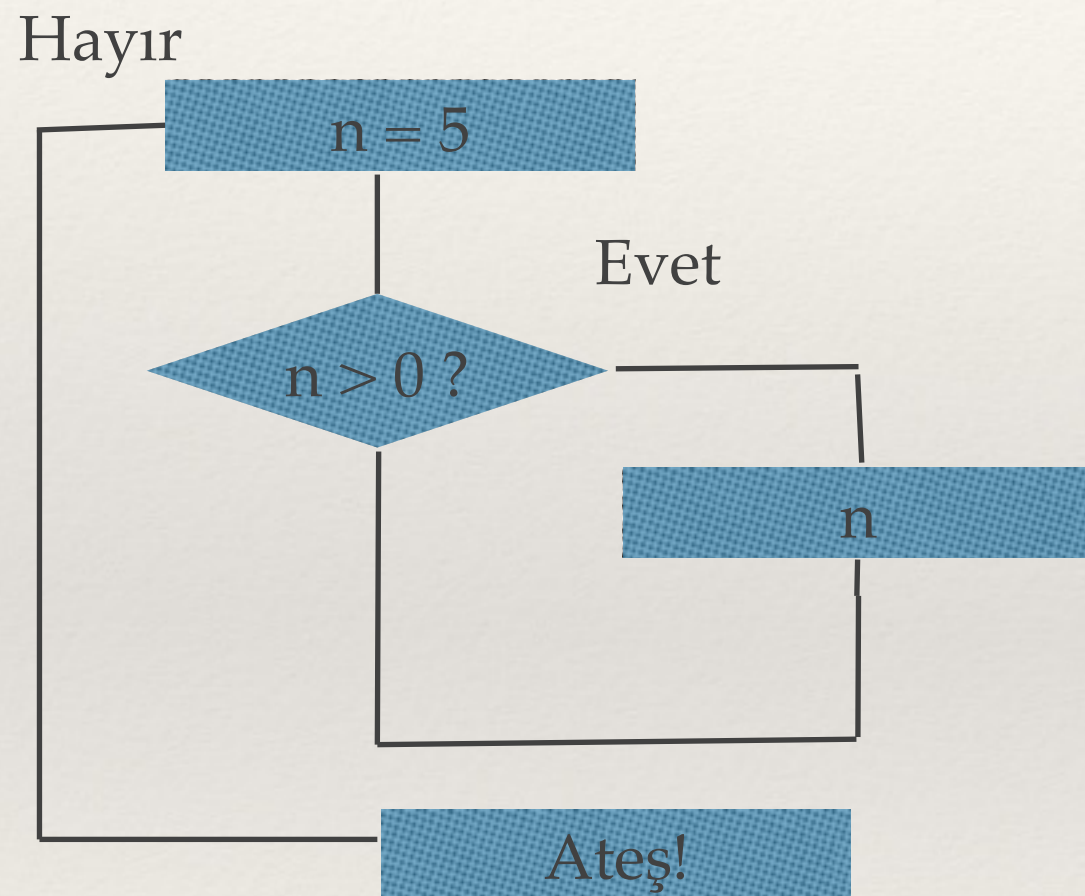
```
#!/python3
n = 5
while n > 0:
    print(n)
    n = n - 1
print("Ateş!")
```

Çıktısı:

5
4
3
2
1
Ateş!

Belirsiz Döngüler

Örnek: Sonsuz Döngü

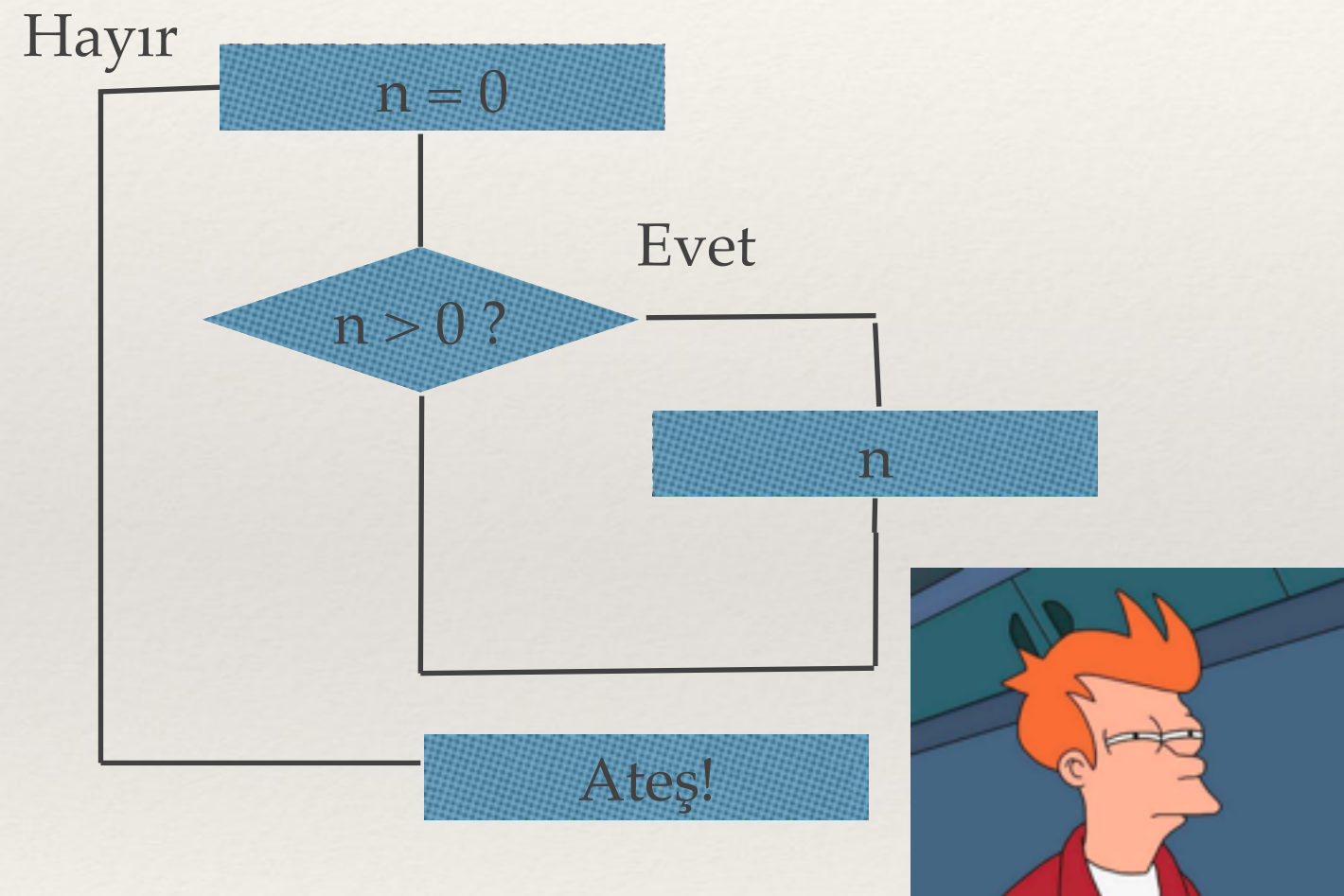


```
#!/python3
n = 5
while n > 0:
    print(n)
    print("Ateş!")
```

Bu döngüdeki sorun nedir?

Belirsiz Döngüler

Örnek: Başka bir döngü



```
#!/python3
n = 0
while n > 0:
    print(n)
    print("Ateş!")
```

Peki bu döngü ne yapıyor??

Döngüden Çıkmak

- ❖ **break** komutu ile döngünün içindeki yerden zıplayarak döngü dışındaki komuta **ışınlanabiliriz** !



```
#!/python3
n =5
while n>0:
    print(n)
    n=n-1
    if n==2:
        break;
print("Ateş!")
```

Çıktısı:

5
4
3

Ateş!

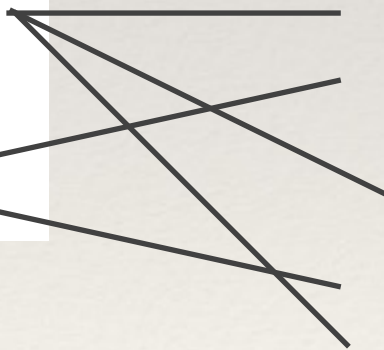
Bir yinelemeyi continue ile bitirmek

- ❖ continue komutu ile geçerli olan döngünün geriye kalan kısmı işleme konulmaz bir sonraki döngüye sıçrar.

```
#!/python3
n =6
while n>0:
    n=n-1
    if n%2==1 :
        print("Tek sayi: ",n)
        continue
    print("Cift sayi:", n)
```

Çıktısı:

5
4
3
2
1



Exit ile programdan çıkmak

- ❖ exit komutu programdan çıkmak için kullanılır. Kullanımındaki amaç özel bir çıkış kodu ile programı sonlandırmaktır.
- ❖ Programın devam etmesini istemediğimiz durumlarda kullanabiliriz.

Exit ile programdan çıkmak

- ❖ exit komutu programdan çıkmak için kullanılır. Kullanımındaki amaç özel bir çıkış kodu ile programı sonlandırmaktır.
- ❖ Programın devam etmesini istemediğimiz durumlarda kullanabiliriz.

```
#!/python3
n =5
while n>0:
    n=n-1
    if n%2==1 :
        exit("Hata! Tek sayi program sona erdiriliyor!")
    print("Cift sayi:", n)
```

Belirli Döngüler

- ❖ Yineleme sayısı belli olan döngülerdir.
- ❖ Döngüde yapılacakları biliyorsak ve yineleme sayısını biliyorsak `for` döngüsünü kullanmamız bize kolaylık sağlayacaktır
- ❖ Python dilindeki `for` döngüsü çok kullanışlıdır. İleri zamanlarda öğreneceğimiz, listeler, diziler gibi verilerin bulunduğu bir yapının içerisinde kolayca döndürebiliriz.
- ❖ Aynı şekilde `range()` fonksiyonu kullanarak belirli bir düzende döngüleri kullanabiliriz.

Belirli Döngüler Örnek:

```
for i in [5, 4, 3, 2, 1] :  
    print(i)  
print( 'Ates!' )
```

- ❖ 5 ten geriye sayıp ateşleme yapan programı for döngüsü ile yazalım.

Belirli Döngüler Örnek Sözcükler:

```
friends = ['Ahmet', 'Ayse', 'Huseyin']  
for friend in friends :  
    print('Mutlu yıllar:', friend)  
print('Bitti!')
```

Mutlu Yıllar: Ahmet
Mutlu Yıllar: Ayse
Mutlu Yıllar: Huseyin

Bitti!

Döngülerin Yapısının İncelenmesi

İterasyon
değişkeni

5 elemanlı dizi

```
for i in [5, 4, 3, 2, 1] :  
    print(i)
```

Döngüler range()

- ❖ Bazen bir liste veya dizi kullanmak yerine belirli sırada sayıları kullanarak hesap yapmak isteyebiliriz.
- ❖ Bu gibi durumlarda range fonksiyonu işimize yarar

```
for i in range(1,5) :  
    print(i)
```

1
2
3
4

Döngüler range()

- ❖ Bazen bir liste veya dizi kullanmak yerine belirli sırada sayıları kullanarak hesap yapmak isteyebiliriz.
- ❖ Bu gibi durumlarda range fonksiyonu işimize yarar

```
for i in range(1,5) :  
    print(i)
```

1
2
3
4

Tersini yapmak istersek?

Döngüler range()

- ❖ Reversed fonksiyonunu range içinde kullandığımızda döngü tersine çevrilir.

```
for i in reversed(range(1,5)) :  
    print(i)
```

4
3
2
1

Tersini yapmak istersek?

Döngüler Farklı İterasyonlar

- ❖ Döngü değişkenini ikişer ikişer veya üçer üçer arttırmak isteyebiliriz. Bu gibi durumlarda range() fonksiyonunun özelliklerini kullanabiliriz.

```
for i in range(0,20,2) :  
    print(i)
```

0
2
.
.
.
18

Döngüler Farklı İterasyonlar

- ❖ Döngü değişkenini ikişer ikişer veya üçer üçer arttırmak isteyebiliriz. Bu gibi durumlarda range() fonksiyonunun özelliklerini kullanabiliriz.

```
for i in range(0, 20, 2) :  
    print(i)
```

Başla Bitir Artım

0
2
.
.
.
.
18

Akıllı Döngü Yazmak

- ❖ Akıllıca döngü yazmak için en önemli **ipucu**, tüm döngü için yapılacak şeyi bildiğimiz halde programı yazarken, her döngü için nasıl çalışacağını öngörmemiz gerekmektedir. Genelde bu kısımda hata yaparız. Bu yüzden!

Bazı değişkenlere başlangıç değeri atamalıyız

for **thing** in data:

Değişkeni değiştirerek her döngü için nasıl değiştiğini takip etmeliyiz

Değişkenlerin son haline bakmalıyız

Dizi boyunca döngü kullanmak

```
print( 'Once' )
```

```
for thing in [9, 41, 12, 3,  
74, 15] :
```

```
    print(thing)
```

```
print( 'Sonra' )
```

```
$ python basicloop.py
```

Once

9

41

12

3

74

15

Sonra

En Büyük Sayı Nedir?

3 41 12 9 74 15

En Büyük Sayı Nedir?

3

En Büyük Sayı Nedir?

41

En Büyük Sayı Nedir?

12

En Büyük Sayı Nedir?

9

En Büyük Sayı Nedir?

74

En Büyük Sayı Nedir?

15

En Büyük Sayı Nedir?

3 41 12 9 74 15

En Büyük Sayı Nedir?

en_buyuk



-1

En Büyük Sayı Nedir?

3

en_buyuk

3

En Büyük Sayı Nedir?

41

en_buyuk

41

En Büyük Sayı Nedir?

12

en_buyuk

41

En Büyük Sayı Nedir?

9

en_buyuk

41

En Büyük Sayı Nedir?

74

en_buyuk

74

En Büyük Sayı Nedir?

en_buyuk

74

15

En Büyük Sayı Nedir?

3 41 12 9 74 15

en_buyuk

74

En Büyük Sayı Nedir?

```
en_buyuk = -1
print('Once', en_buyuk)
for sayi in [9, 41, 12, 3, 74, 15] :
    if sayi > en_buyuk :
        en_buyuk = sayi
        print(en_buyuk, sayi)
print('Sonra', en_buyuk)
```

\$ python largest.py

Once -1

9 9

41 41

41 12

41 3

74 74

74 15

Sonra 74

Acknowledgements / Contributions

These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and open.umich.edu and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors and Translators here

