Python Programlama

Yazılım Mühendisliği Dersleri İçin Hızlı Python Klavuzu

Dr. Yunus Santur

Kısaca

- Python
- 1990 yılında geliştirildi
- C temelli
- Guido Van Rossum (Baş Geliştirici)
- Açık kaynak
- Cross-platfrom
- Öğrenmesi kolay: «Hayat kısa python öğrenin»
- Yaygın kullanım

Genel özellikler

- Pythonda Olmayanlar
  - Küme parantezi
  - Begin-end benzeri ifadeler
  - satır sonu ; kuralı
  - Değişken tipi tanımlama

- Girintileme kuralı var!

Python Etkileşimli Kabuk

Python yüklü ise (veya **Anaconda** ortamı)

Komut satırından python Shell çalıştırılır. **>>>**

Online python kabuğu https://www.python.org/shell/

```
>>> a=2    # Değişken tanımlama
>>> a**2  # Karesini alma
>>> help(a) # Yardım dökümanı
>>> dir(a) # Herhangi değişken yada metotla ilgili tüm seçenekler
>>> a.bit_length() # Değişkenin kaç bit kapladığı
>>> a.__abs__() # Sayının mutlak değerini alma

>>> type (değişken) # Değişkenin tipi nedir ?
```

Girintileme Kuralı

Metot/Şart/Döngü Kodlarından sonra (blok)**:** koymak girintileme yapmak kuraldır (editör ortamında tab tuşu ile)

Ana Program

Blok İfadesi:

    Bloka ait kodlar

    …

Ana Program Akışı

Yorum satırları

ile başlayarak tek satırlık yorum satırları oluşturulabilir.

Birden fazla satırı yorum satırı haline getirmek için

```
" " "
 Yorum satırı 1
 Yorum satırı 2
 devam…
" " "
```

Aritmetik İşleçler

+

-

*

**

/

//

%

+=

-=

*=

/=

Karşılaştırma İşleçleri

==	eşittir
!=	eşit değildir
>	büyüktür
<	küçüktür
>=	büyük eşittir
<=	küçük eşittir

Bool İşleçler

İki yada daha fazla şartı birleştirme

and

or

not

"p" in "Python" => False

a is 256 => True

Şart İfadeleri

if *boolean_ifade*:

    if şartına ait kodlar

if boolean_ifade:

        şart ifadesi doğru ise

else:

        şart ifadesi yanlış ise

Kısa if

```
if şart:
        doğru_ise
else:
        yanlış_ise
```

- **Tek satırda yazma**

```
doğru_ise if şart else yanlış_ise
```

if-elif-if

```
if şart:
        şart1 doğru ise
elif şart:
        şart2 doğru ise
elif şart:
        şart3 doğru ise

…

else:
        default kısım
```

if-else örneği

```
a=2
b=3
if a<b:
        print "a b'den küçük"
else:
        print "a b'den büyük"
```

if-elif-else örneği

```
puan=36
if puan>85:
        print "A "
elif puan>70:
        print "B "
elif puan>50:
        print "C "
elif puan>40:
        print "D "
else:
        print "F "
print " \n "
```

Veri Tipleri

- Temel tipler (integer, string, boolean,double)
- *Değişken tipi belirtilmez!*

- Diğer veri tipleri
  - Liste (List)
  - Demet (Tuple)
  - Küme (Set)
  - Sözlük (Dictionary)

Listeler

```
a=[1,2,"a",3,5]        # Veriler karışık türden olabilir

len(a)                 # Eleman sayısı
a.sort()               # listeyi sıralar
a.reverse()            # listeyi ters çevirir
a.pop()                # son elemanı siler
a.append("a")          # sonuna yeni eleman ekler
a.insert(indis, "a")   # yeni elemanı belirtilen indise ekler
```

Listeler devam

```
a.count(1)        # Bu eleman listede kaç tane var
a.index(1)        # Bu eleman kaçıncı indiste

print a[1]        # 1.indiste ki elemanı yazdır
a[1]=2            # 1.indisteki elemanın değerini değiştir
del a[2]          # 2.indisteki elemanı listeden sil

x=list()          # Boş liste oluşturur
x=[]              # Boş liste oluşturur
```

Listeler Devam

Dahası ?

```
>>> dir(a)              # a ile başka ne yapabilirim
```

Kabukta belirli bir fonksiyonla ilgili yardım alma

```
>>> help(a.append)      # a.append nasıl çalışır
```

Demetler

Listelere benzerler!

a=(1,2,"a")

a.count(1) # 1 elemanı kaç tane var

a.index(2) # Bu elemanın index'i

print a[1] # 1.elemanını yazdır

Demetler tanımlandıktan sonra güncellenemezler!

Kümeler

a={1,2,"a",5}

#Bu methodları desteklerler
add
revome
pop

#Ancak aşağıdaki methodlar çalışmaz, kümeler sıralıdır ve çift değer içermezler
a.index
a[1]
a.count(5) ?

Sözlükler

x={"isim":"ali", "meslek":"muhendis","maas":1000, "ehliyet":True}

{key:value, key:value, key:value …}

print x[key]=value # Değeri yazdırma
x[key]=new_value # Değeri değiştirme
x[new_key]=value # Yeni key:value çifti ekleme

x.keys() # x'in anahtarları
x.values() # x'in değerleri

Döngüler

<u>For Döngüsü Yapısı:</u>

```
for i in range(a,b,c):
        print i
```

a’dan b’ye (a dahil b değil), c artımlı döngü
range(10): 0…9
range(2,10): 2 3 4 5 6 7 8 9
range(2,10,3): 2 5 8

For Döngüsü

break

döngüyü kırıp bitirir, iç içe döngülerde sadece ait olduğu iç döngüyü bitirir

continue

döngüyü pas geçer (bir sonraki adımdan devam eder)

Liste, Demet, Set Üzerinde Döngüler

```
list=[1,2,5,10]
for i in range(0,len(list)):
        print list[i]
```

Pythonic versiyon

```
list=[1,2,5,10]
for i in list:
        print i
```

Sözlük Üzerinde Döngüler

```
x={"isim":"ali", "meslek":"muhendis","maas":1000, "ehliyet":True}

for (k,v) in x.items():
        print k,":",v
```

k: sözlük anahtarlarını

v: sözlük değerlerini simgeliyor

While Döngüsü

while *şart_ifadesi*:

kodlar

Dikkat:

Blok içerisinde döngü artımı/döngüden çıkış şartı olmazsa sonsuz döngü yapmış oluruz

Metotlar

```
def metot_ismi(parametre_listesi):
        metot_kodları
        return değer
```

Metotlar devam

```
def hesapla(a,b):
        x=a*b
        return x


print hesapla(3,4)
```

Metotlar devam

```
def hesapla(a=2,b=3):
        x=a*b
        return x



print hesapla()         # Sonuç 6 olacak
print hesapla(5)       #  Sonuç ?
# Metot parametresiz çağrılırsa default değerler alınır
```

Metotlara belirsiz sayıda parametre göndermek

```
def hesapla(*liste):
        t=0
        for i in liste:
                t+=i
        return t


print hesapla(1,2,3,4,5,6)
```

Modul Kullanımı

Moduller kütüphanelerdir

```
import modül_ismi       # Modül programa dahil edilmiş olur
modül_ismi.method       # Modülün methodunu kullanmak

import string           # String modülü
import random           # Random modülü
```

Random Modülüne Giriş

```
random.randint(a,b)        # a-b aralığında bir tam sayı tutar
random.random()            # 0-1 aralığında bir rasyonel sayı tutar
```

```
x=[1,2,5,10]
print random.choice(x)          # x listesinden rastgele bir eleman seçer
print random.sapmle(x,3)        # x listesinden rastgele üç eleman seçer
```

String İfadeler

```
s="Merhaba Python"
print s
```

String ifadeler liste veri yapısına çok benzerler, örneğin len(s) karakter sayısını verir

```
s.upper()               # Büyük harfe dönüştür
s.lower()               # Küçük harfe dönüştür
s.count("python")    # python kelimesi kaç defa geçiyor

s[3].isupper()          # Stringin 3.indisteki karakteri büyük harf mi ?
s[3].islower()          # Stringin 3.indisteki karakteri küçük harf mi ?
s[7].isdigit()          # Stringin 7.indisteki karakteri rakam mı ?

x=s.split("")                              # Boşluk karakterine göre ayırarak listeye atar
s1=''.join(random.sample(s,3))          # s Stringinden rastgele 3 karakteri al s1 Stringinde birleştir
```

join ve split birbirlerinin tam tersi iş yaparlar

Listeleri Birleştirmek

```
a=[1,2,3]
b=[4,5,6]
c=a+b
```

Çok Boyutlu Listeler

```
x=[
    [1,0,0],
    [0,1,0],
    [0,0,1]
  ]

len(x)        # satır sayısını verir
len(x[0])    # ilk satırdaki sütun sayını verir
```

Çok Boyutlu Listeler

```
x=[
    [1,0,0],
    [0,1,0],
    [0,0,1]
  ]

for i in x:
        print i
```

Çok Boyutlu Diziler

```
import random
x=[]
for i in range(5):
        x.append([random.randint(1,5) for c in range(4)])


print x
```

Döngülerle Birlikte Else Kullanımı

```
while şart:
        komutlar1
else:
        komutlar2
```

Döngü yanlış ise else kısmı çalışacak

```
while şart:
        komutlar1
else:
        komutlar2
```

Klasik yaklaşımda ki boolean tipte bir kontrol değişkeni kullanımı gereksiz olur

For-Else Örneği

- **Klasik yöntem**

```
x=25
asal=True
for i in range(2,x/2+1):
        if x%i==0:
                asal=False
                break
if asal:
        print x,"asal bir sayidir»
else:
        print x,"asal degildir"
```

- **For-Else ile**

```
x=25
for i in range(2,x/2+1):
        if x%i==0:
                print x,"asal sayi değildir"
                break
else:
        print x,"asal bir sayidir"
```

Liste Üreteçleri

- **Klasik yaklaşım**

```
x=[]
for i in range(100):
        x.append(i)

print x
```

- **Pythonic**

```
x=range(100)

print x
```

Liste Üreteçleri

- **Klasik yaklaşım**

```
x=[]
for i in range(100):
        x.append(i**2)

print x
```

- **Pythonic**

```
x=[i**2 for i in range(100)]

print x
```

Her iki programda
1,4,9,25,36 … n^2
den oluşan bir liste üretir

Satır İçi Fonksiyonlar

```
x=list()
x=[i**2 for i in range(100)]
```

- **Biraz Daha İleri Seviye**

```
str="Hayat Kısa Python Oğrenin"
x=[ i for i in str if i.isupper() ]
```

X ne olur ?