BİLGİSAYAR BİLİMİ DERSİ DERS NOTLARI

DEĞİŞKENLER

Değişkenler veri tutan ve saklayan birimlerdir
1- Değişkenlere Tek Tek değer atayabiliriz
Ad=”İsmail”
Soyad=”YALÇIN”

2-Değişkenlere Sıra ile değer atayabiliriz
Ad,soyad,sinif,yas=”Ahmet”,”SOYLU”,”9K”,”15”

3-Birden fazla değişkene aynı değeri atayabiliriz
Elma=kiraz=erik=3

VERİ TİPLERİ

Şimdiye kadar 3 veri tipi gördük

VERİ TİPİ	AÇIKLMA	ÖRNEK
String veri tipi	Metin , yazı ve karakter dizileri string veri tipine örnektir	“Ahmet” “Anibal” “54kljlişlşi87” “5056464656”
İnteger veri tipi	Tam sayılar bu veri tipine aittir	23 5 656556456
Float veri tipi	Noktalı sayılar bu veri tipine aittir	3.14 6.545465 0.235

type () fonksiyonu

Veri tiplerini öğrenmek için kullanılır

Örnek:

parola=”kljals@uhd’1+?”
sayi1=65
pi=3.14
type(parola)-----→ string değerini verir
type(sayi1)------→int değerini verir
type(pi)----------→float değerini verir

print( ) Fonksiyonu

Program çalıştığı anda içeriklerin gösterilmesini yazda yazdırılmasını sağlar
Örnek
İsim=”Ahmet”
print(“hoş geldin ”, isim)

yukarıdaki örnek “ hoş geldin Ahmet “ yazar

len( ) fonksiyonu

String bir verideki karakterlerin sayısını verir
ÖRNEK
İsim=”Kadir”
karakterSayisi=len(isim)
print(karakterSayısı)

sonuç olarak “merhaba” string bir veridir ve karakter sayısı=7 dir

del() Fonksiyonu

Değişkenleri silmek için kullanılır
ad=”Ahmet”
Soyad=”DİNÇER”
del(ad)
del(soyad)

Yukarıdaki örnekte ad ve soyad değişkenleri, silinmiştir

Matematiksel oparatörler

Matematiksel işlemleri yaparken kullanılan oparatörler aşağıdaki tabloda gösterilmiştir

Toplama	+	Toplam=a+b
Çıkarma	-	Cikarma=a-b
Çarpma	*	Çarpma=a*b
Bölme	/	Bolme=a/b
Us Alama	**	usAl=a**b
Mod alma	%	modAl=a%b

İnput() Fonksiyonu

Bu fonksiyon ile program çalıştığında veri girişi yapılmasını sağlar. Girilen veriler string veri tipindedir. Girilen veriler üzerinde sayısal işlem yapmak için tam sayıya =integer(int) veya noktalı sayı=float veri tipine dönüştürülmelidir

ÖRNEK1:Girilen isme göre işlem yapan kodlar aşağıdaki gibidir.

```
isim=input("İsminizi giriniz")
print("Hoş geldin ",isim)
```

Bu örnekte Program çalıştırıldığında " isminizi giriniz " mesajı çıkacak ve imleç yanıp sönecek . İsim girildikten sonra(mesala "Ayşe") print fonksiyonu da isim değişkeninin tuttuğu veriyi "Hoş geldin Ayşe" gibi yazdıracaktır

ÖRNEK2 Girilen a ve b sayısını toplayan kodlar aşağıdaki gibidir. Bu örnekte girilen değerler üzerinde sayısal işlem yapılacağından int() fonksiyonu ile tam sayıya dönüşüm yapılmıştır.

```
A=int ( input ( "A sayısını giriniz:" ) )
B=int ( input ("B sayısını giriniz:") )
Topla=A+B
print(" toplama işleminin sonucu =" ,Topla)
```

Algoritmalar

Program yazmaya başlamadan önce yapılacak işlemlerin adım adım yazılmasına «algorima» denir

ÖRNEK: Dikdörtgenin çevresini bulan programın algoritmasını yazınız

1.adım:başla
2.adım:a ve b kenarlarının değerini gir
3.adım:çevreyi hesapla>> cevre=2*(a+b)
4.adım:çevreyi göster-yazdır
5.adım:bitir

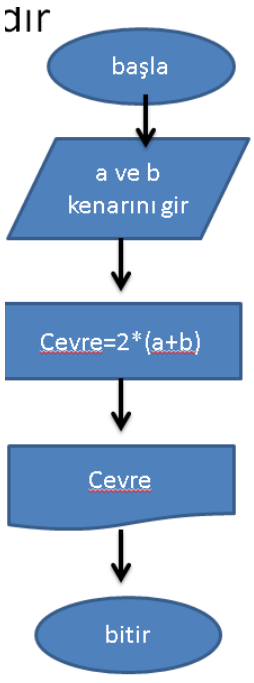
İş Akış Şeması

Akış şeması ise algoritmanın görsel gösterimidir. Akış şeması ile programın çalışma yapısı daha anlaşılır hale gelecektir.

Simge	İşlev
	Başla/Bitir
	Giriş
	Atama/İşlem
	Denetim (Karar)
	Çıkış
	Döngü
	Akış Yönü
	Bağlaç
	Önceden Tanımlı İşlem/Fonksiyon

Şekil 1.10: Akış şeması sembolleri

dır



KARŞILAŞTIRMA OPARATÖRLERİ

Python'da verileri karşılaştırmak için karşılaştırma operatörleri kullanılır.

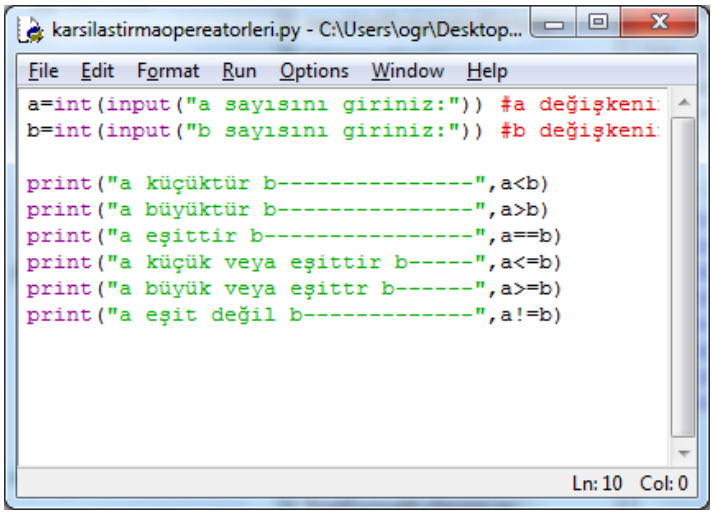
Operatör	Anlamı	Örnek	Açıklama
==	Eşittir	x==y	X değeri y değerine eşitse Sonuç doğru(true) değilse Yanlıştır(False)
<	Küçüktür	x<y	X değeri y değerinden küçükse sonuç doğru (true) değilse Yanlıştır (false)
>	Büyüktür	x>y	X değeri y değerinden büyükse sonuç doğru (true) değilse Yanlıştır (false)
<=	Küçük veya eşittir	x<=y	X değeri y değerinden küçük veya eşitse sonuç doğru(true) değilse yanlıştır(false)
>=	Büyük veya eşittir	x>=y	X değeri y değerinden büyük veya eşitse sonuç doğru(true) değilse yanlıştır(false)
!=	Eşit değildir?	x!=y	X değeri y değeri ile eşit değilse Sonuç Doğru(true) eşitse yanlıştır(false)

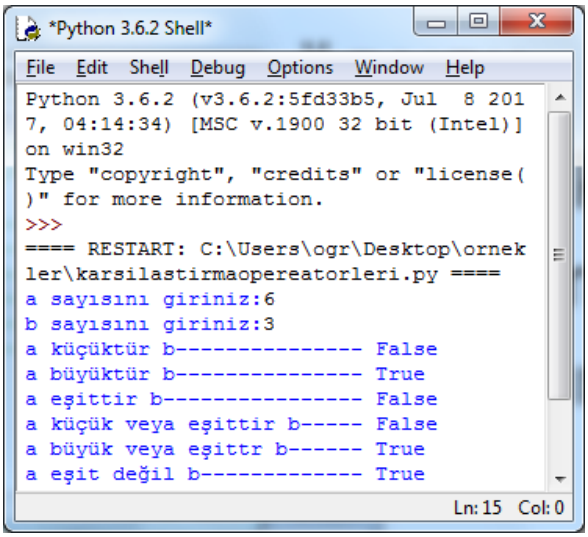
ÖRNEK

Operatör	Anlamı	Örnek	sonuç
==	Eşittir	6==6	Doğru-True
<	Küçüktür	3<2	Yanlış-false
>	Büyüktür	8>5	Doğru-true
<=	Küçük veya eşittir	3<=3	Doğru-true
>=	Büyük veya eşittir	6>=7	Yanlış-False
!=	Eşit değil	4!=3	Doğru-True

ÖRNEK

Girilen a ve b sayılarını karşılaştıralım ve önermelerin doğru mu, yanlış mı olduğunu yazdıralım





İf ,else Koşul ifadelerinin kullanılması

Türkçede EĞER anlamına gelen if ifadesi, adından da anlaşılacağı üzere, koşula bağlı durumları kontrol etmek amacıyla kullanılır.

Koşul doğru(true) ise if altındaki satırlar çalışır

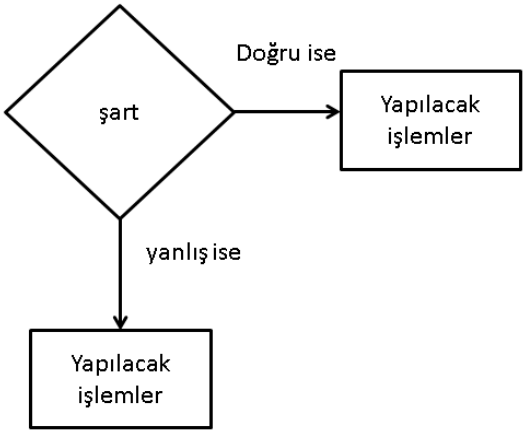
Koşul yanlış(false) ise else altındaki satırlar çalışır

İf Koşul:

koşul doğru(true) ise yapılacak işlemler

else:

Koşul yanlış(false) ise yapılacak işlemler



Girilen sayının 5 den küçük olup olmama durumunu gösteren programın algoritma, akış şeması ve python kodlarını yazalım

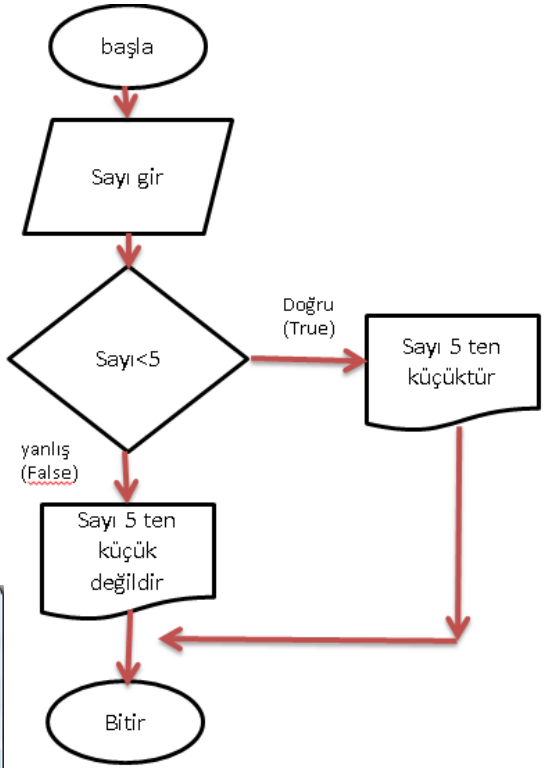
1.adım: Başla

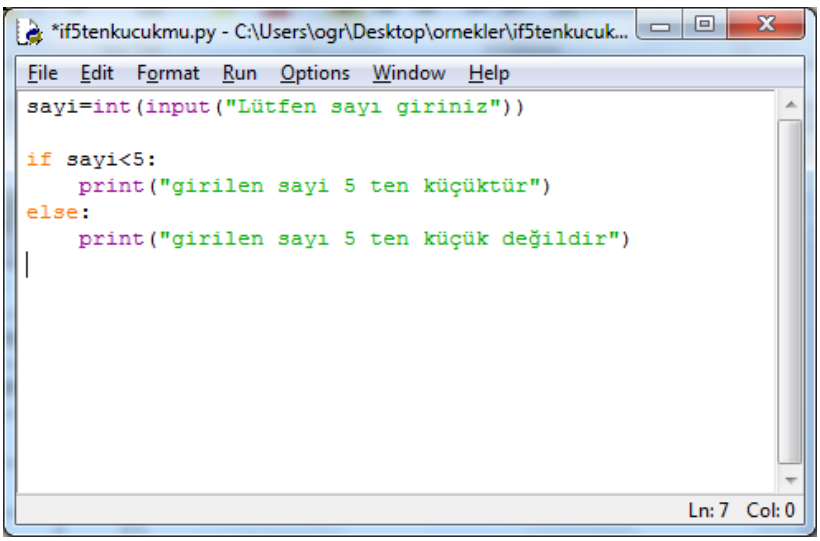
2.adım: Sayı gir

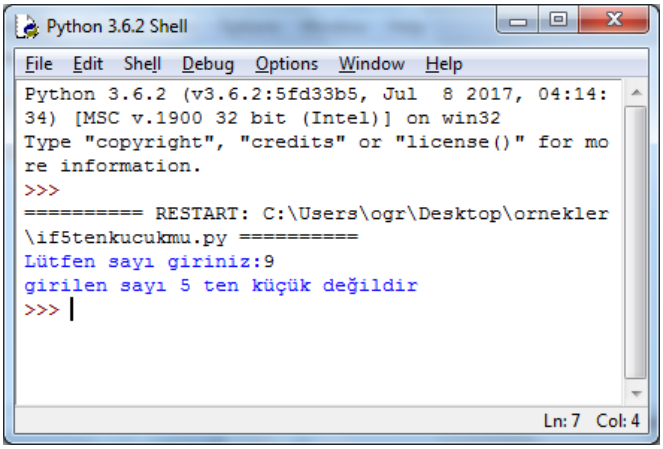
3.adım: Eğer sayı 5 ten küçükse « girilen sayı 5ten küçüktür yaz»

değilse «girilen sayı 5 ten küçük değildir» yaz

4.adım:Bitir







İf, elif, else Koşul İfadelerinin Kullanımı

İf ve else kullanımını örnekler ile gördük. Eğer veri üzerinde birden fazla şart kullanılacaksa elif fonksiyonu kullanılmalıdır.

if şart1:

    şart 1 doğru ise yapılacaklar

elif şart2:

    şart 2 doğru ise yapılacaklar

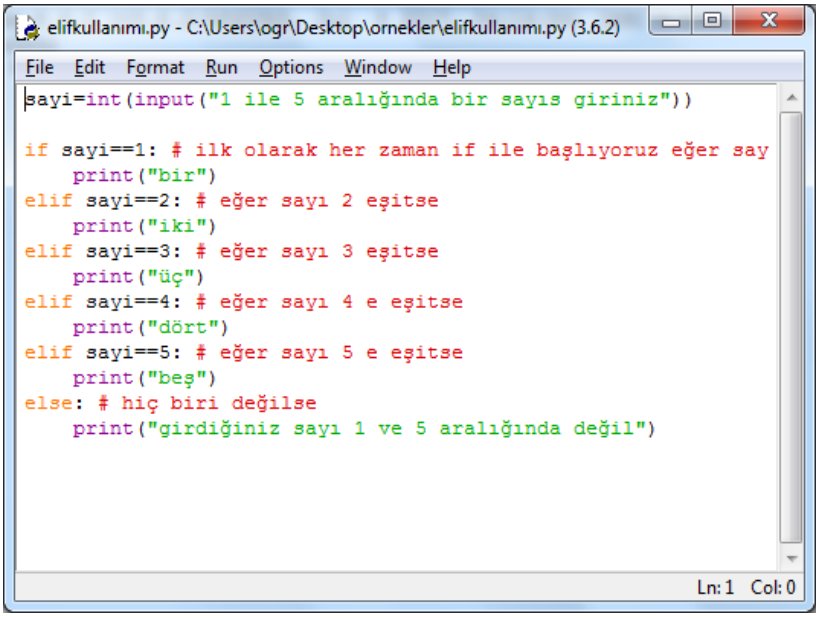
elif şart3:

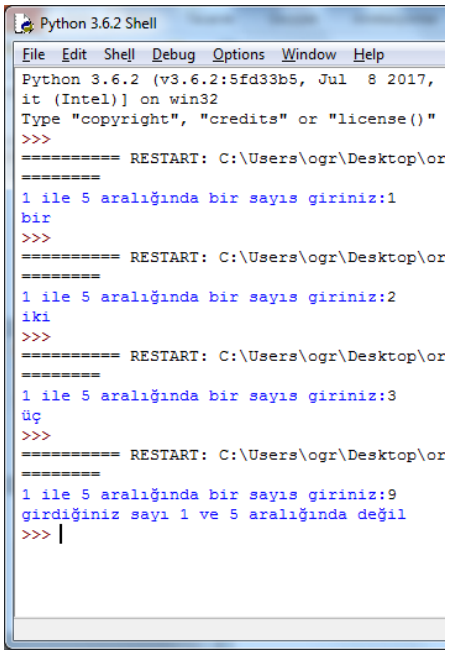
    şart 3 doğru ise yapılacaklar

else:

    Hiçbir şart uygun değilse yapılacaklar

ÖRNEK:1 den 5 e kadar girilen sayıların yazılışlarını gösteren programı yazınız





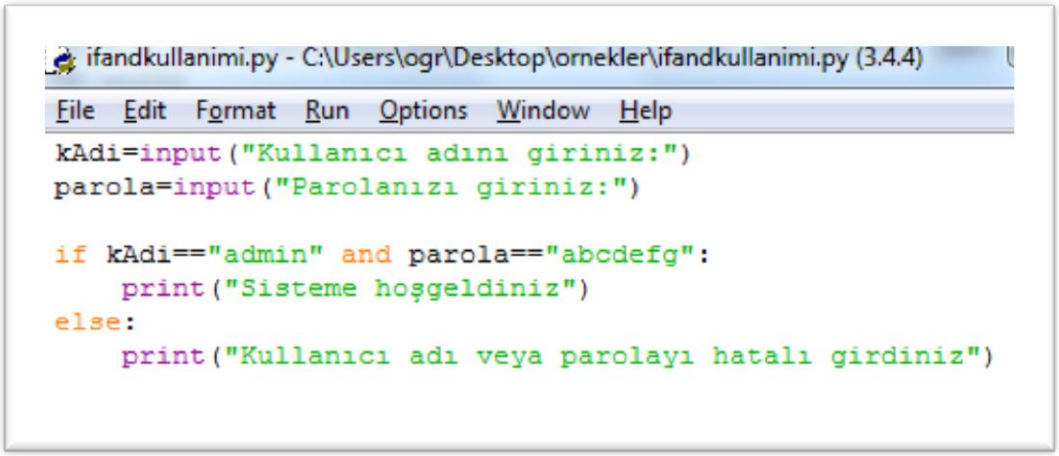
KOŞULLU İFADELERDE AND, OR, NOT ve İN KULLANIMI

AND KULLANIMI

«And» ifadesi «ve» anlamına gelir. Koşullu ifadelerde birden fazla şartın doğru(true) olması durumunda işlem yapılması isteniyor ise and işleci kullanılmalıdır.

```
if şart1 and şart2:
        Tüm şartların  doğru(true) olması durumunda yapılacak işlemler
else:
        şartların birinin veya tümünün yanlış(false) olma durumunda yapılacak işlemler
```

ÖRNEK:Bir hesaba kullanıcı adı ve parola ile giriş yapılacaktır. Eğer Kullanıcı adı «admin» ve parola «abcdefg» ise siteme giriş yapılacaktır. değilse, «kullanıcı adınız veya parolanız yanlış» çıktısını veren kodları yazınız.



«And» işlecinin mantıksal gösterimi

İf şart1 and şart2:

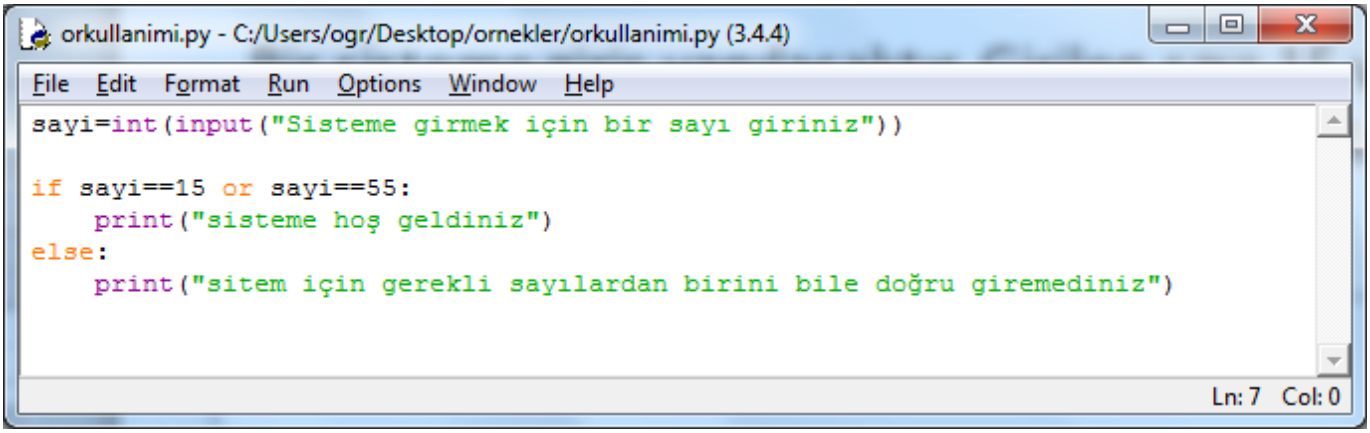
Şart1	şart2	sonuç	İşletilecek yer
True	True	True	İf altındaki satırlar
True	False	False	Else nin altındaki satırlar
False	True	False	Else nin altındaki satırlar
False	False	false	Else nin altındaki satırlar

OR(VEYA) KULLANIMI

«or» ifadesi «veya» anlamına gelir. Koşullu ifadelerde birden fazla şarttan sadece birinin doğru(true) olması durumunda işlem yapılması isteniyor ise or işleci kullanılmalıdır.

```
if şart1 or şart2:
        şartlardan biri  doğru(true) olması durumunda yapılacak işlemler
else:
        şartların tümünün yanlış(false) olma durumunda yapılacak işlemler
```

ÖRNEK: Bir sisteme giriş yapılacaktır. Girilen sayı 15 veya 55 ise «sisteme kabul edildiniz» değilse «giriş yetkiniz yok» yazan programı oluşturunuz



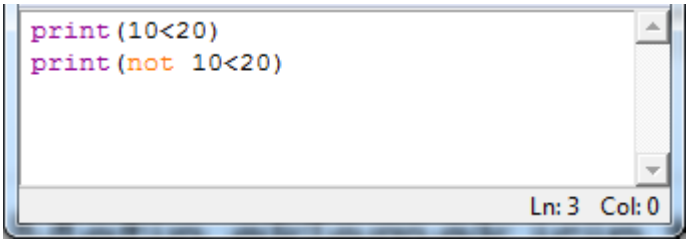
«or» işlecinin mantıksal gösterimi

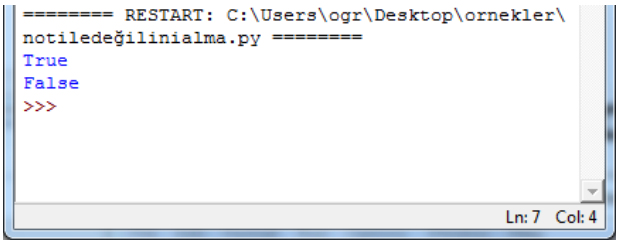
İf şart1 orşart2:

Şart1	şart2	sonuç	İşletilecek yer
True	True	True	İf altındaki satırlar
True	False	True	İf altındaki satırlar
False	True	true	İf altındaki satırlar
False	False	false	Else nin altındaki satırlar

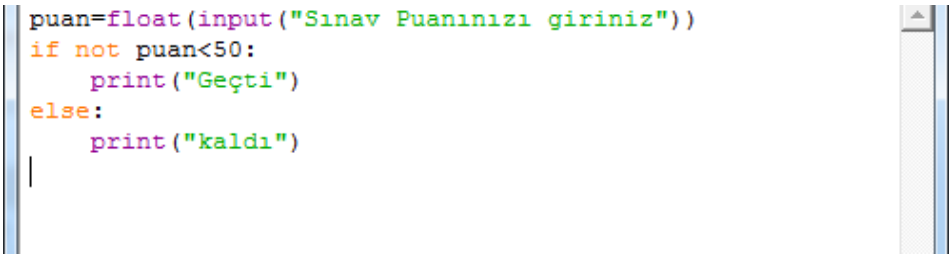
NOT(DEĞİL) KULLANIMI

- Not Türkçede değil anlamına gelir. Bu işleç, true değeri False yaparken False bir durumu true yapar
- Aynı zamanda özellikle kullanıcı tarafından bir değişkene veri girilip girilmediğini denetlemek için kullanılabilir.





ÖRNEK:Girilen puan 50 den küçük değilse geçti yazan programı oluşturalım

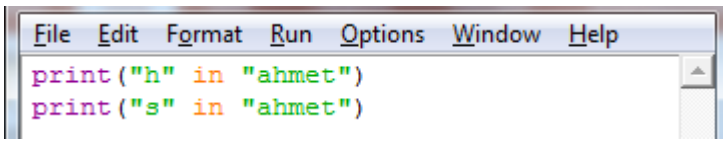


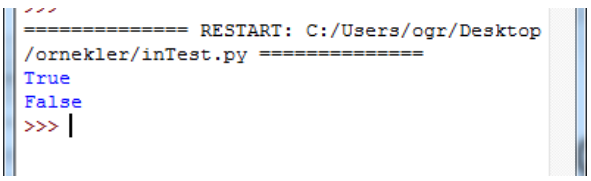
not (değil) işlecinin mantıksal göstesimi

Şart	not şart
True	False
False	True

İN (İÇİNDE) AİTLİK İŞLECİNİN KULLANILMASI

- Aitlik işleçleri, bir karakter dizisi ya da sayının, herhangi bir verinin içinde bulunup bulunmadığını sorgulamamızı sağlayan işleçlerdir. Varsa **True** yoksa **False** sonucunu verir.





ÖRNEK: Girilen bir email adresi içerisinde «@» işareti varsa «email adresiniz geçerlidir» yoksa «Email adresiniz hatalıdır» yazan. Python kodlarını yazınız

```
email=input("Lütfen email adresisnizi giriniz:")

if "@" in email:
    print("Email adresiniz geçerlidir")
else:
    print("email adresinizde @ işareti olmalı")
```

ÖRNEK: Girilen harf sesli mi, değil mi? kontrolünü yapan programı yazalım

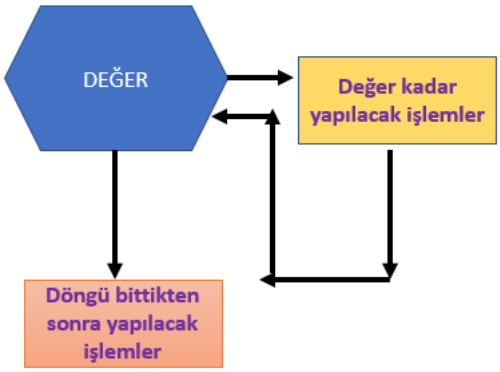
```
sesliHarfler="aıoüeiö"
harf=input("Bir karakter giriniz")
if harf in sesliHarfler:
     print("Girdiğiniz karakter sesli harftir")
else:
    print("Giediğiniz karakter sesli harf değildir")

#Açıklama
#1. satırda SesliHarfler değişkenine sesli harflerimiz atandı
#2.satırda Kullanıcının karakter girişi yapması isteniyor ve harf değişkenine atanıyor
#3.satırta eğer alınan harf sesli harflerin içinde var ise ...
#       Gidiğiniz karakter sesli harfti cevabı yazdırılıyor
#değilse
#       Girdiğiniz karakter seli harf değildir yazdırılıyor
```

FOR DÖNGÜSÜNÜN KULLANILMASI

Döngü nedir?

Döngüler, sıralı bir kod bloğunun istenilen sayıda tekrarlanmasıdır.

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 FONKSİYONUNUN ÜÇ KULLANIMI VARDIR.

range(değer)

range(başlangıç, bitiş)

range(başlangıç, bitiş, artış değeri)

Range Kullanımı	sonuç	Tanımlama
`for sayi in range(5):` `  print(sayi)`	0,1,2,3,4	**range(değer)** 0 dan başlar, değerin 1 eksiğine kadar sayı üretir. Bu döngü 5 kez- defa döner
`for i in range(1,11):` `  print(i)`	1,2,3,4,5,6,7,8,9,10	**range(başlangıç, bitiş)** başlangıç değeri ile başlar, bitiş değerinin 1 eksiğine kadar döngü devam eder.1 den 10 'a kadar sayıları yazar. 11 dahil değildir.
`for sayi in range(5,20,3):` `  print(sayi)`	5,8,11,14,17	range(başlangıç, bitiş, artış değeri) Kullanımı başlangıç değerinden ,bitiş değerinin 1 eksiğine kadar, artış değerince döngü devam eder

FOR DÖNGÜSÜ İLE İLGİLİ ÖRNEKLER

10 defa "Anibal" yazan programı oluşturalım	`for i in range(10):` `  print("anibal")`

girilen ismi , girilen sayı kadar (defa) yazdıran program	`isim=input(" isim giriniz")`
	`kez=int(input("kaç defa yazılsın"))`
	
	`for x in range(kez):`
	`  print(isim)`

50 den 100'e kadar sayıları yazdıralım	`for sayi in range(50,101):` `  print(sayi)`
0,100 e kadar 5 er artarak yazdıran programın kodlarını yazalım	`for sayi in range(0,100,5):` `  print(sayi)`

YENİ ATAMA İŞLEÇLERİ

Atama İşleçleri	**isim**	**Sayi=10**	**Sayi değişkeninin yeni değeri**
=	Ata	Sayi=10	10
+=	Topla ve Ata	Sayi+=5	15
-=	Çıkar ve Ata	Sayi-=5	5
*=	Çarp ve Ata	Sayi*=3	30
/=	Böl ve Ata	Sayi/=2	5

```
sayi=10
print(sayi) # sayı değeri 10 oldu
sayi+=5
print(sayi) #sayi değeri 15 oldu
sayi+=20
print(sayi) #sayi değeri 35 oldu
sayi-=10
print(sayi) #sayi değeri 25 oldu
sayi*=2
print(sayi) #sayi değeri 50 oldu
sayi/=10
print(sayi) #sayi değeri 5 oldu
```

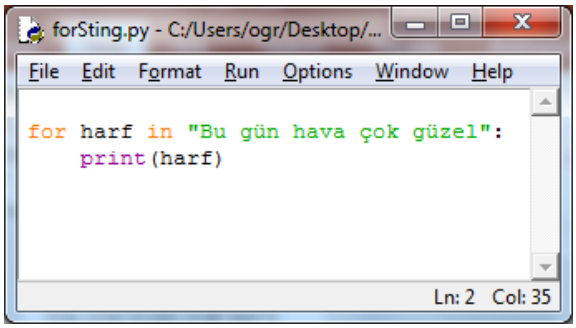
Örnek : For döngüsünü kullanarak 1 den 100'e kadar(100 dahil) sayıların toplamını bulalım

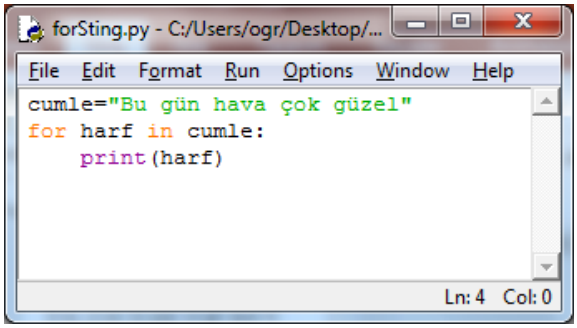
```
toplam=0 # döngüden gelen sayılar toplam değişkenin üzerine eklenecek

for sayi in range(1,101):
    toplam+=sayi  # bu satırda sayılar toplam değişkeninin üzerine ekleniyor

print("Toplama işleminin sonucu:",toplam)
```

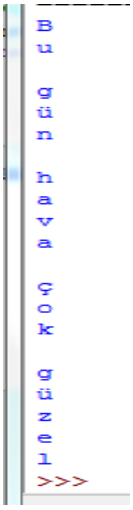
FOR DÖNGÜSÜNDE STRİNG VERİLERİN KULLANILMASI

For döngüsü ile string veri içerisindeki her bir karakteri alabiliriz

Aşağıdaki örnek ile belirtilen tring ifade döngü içerisine adındı ve ker hrf tek tek alınarak yazdırıldı







Örnek: Girilen cümle içerisindeki sesli harflerin sayısını bulalım

```
sesliHarfler="aıoüeiö"
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 değişkeniine 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 içeriisindeki karakterler sıra ile harf değişkenine alınacaktır
#5.satırta eğer alınan harf seslli 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
```

DÖNGÜYÜ BREAK KOMUTU İLE BİTİRME-KESME

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

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

```
for sayi in range(1,11):
    print(sayi)
    if sayi==5:
        break
```

```
1
2
3
4
5
```

Döngü normalde 1,2,3,4,5,6,7,8,9,10 sayılarını yazacakken if sayi==5: koşu ifadesi ile döngü sayi değişkenine 5 değerini atatdıktan sonra break komutu çalışıyor ve döngü sonlandırılıyor

Örnek2) 1000 defa dönen döngü içerisinde girilen kelime «çıkış» ise döngüyü sonlandıran kodları yazalım

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

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

öRNEK 3)1000 defa işlem yapmamıza izin veren döngüyü klavyeden girilen seçenek ile istediğimiz zaman bitirmek

```
for i in range(1000):  # for döngüsü 1000 defa çalışacak
    print("""
    -----İşlemler-----
    1-toplama
    2-çıkarma
    3-ÇIKIŞ
    """)
    secim=int(input("yapmak istediğiniz işlemi seçiniz:"))
    if secim==1:
        sayi1=int(input("1.sayıyı giriniz"))
        sayi2=int(input("2.sayıyı giriniz"))
        toplam=sayi1+sayi2
        print("Toplama işleminin sonucu=",toplam)
    elif secim==2:
        sayi1=int(input("1.sayıyı giriniz"))
        sayi2=int(input("2.sayıyı giriniz"))
        fark=sayi1-sayi2
        print("Çıkarma işleminin sonucu=",fark)
    elif secim==3:  #secim 3 ise döngü sona erecek
        break
    else:
        print("yaptığınız seçimde işlem yoktur"
```

Daha önce menü içerisinden yapılan seçime göre sadece 1 defa işlem yapabiliyorduk. Yandaki programda for döngüsü ile bu hakkımızı 1000 defa yapabiliriz. Aynı zamanda break komutu ile döngüyü durdurabiliriz

FONKSİYONLAR

Fonksiyon Nedir?

Fonksiyonlar, belirli işlemleri gerçekleştiren kod bloğudur.

Python içerisinde yer alan çok fazla sayıda hazır fonksiyon vardır. print(), input(), len(),del() gibi fonksiyonlar bunlardan bir kaçıdır.

Bu dersimizde Python'da kendi fonksiyonlarımızı oluşturup kullanacağız.

Fonksiyon Tanımlama

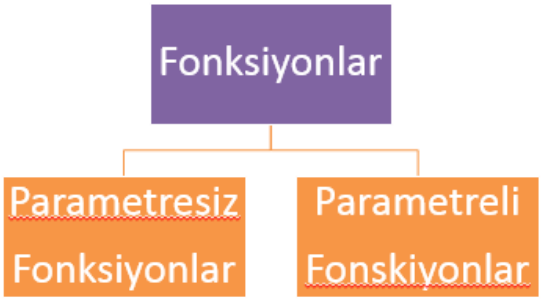
Fonksiyonları program içerisinde kullanabilmek için iki aşama vardır.

1-Fonksiyon tanımlanmalı

2-Fonksiyon çağrılmalı

Fonksiyonlar ikiye ayrılır

FONKSİYONLAR İKİYE AYRILIR

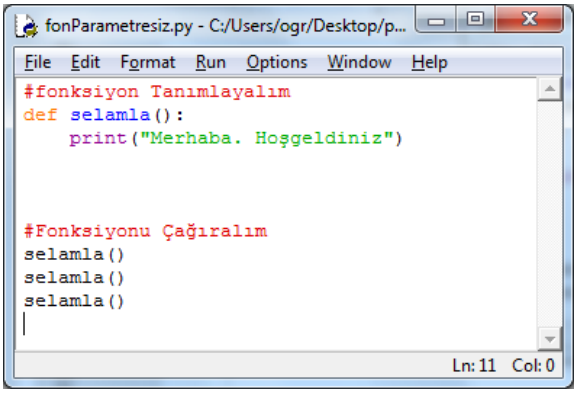


PARAMETRESİZ FONKSİYONLARIN KULLANIMI

Bu tür fonksiyonları tanımlarken fonksiyon ismi yanındaki parantez içerisine parametre girilmez. Fonksiyon ,içerisinde yer alan komutları icra eder

```
def fonksiyon_ismi( ):
        Fonksiyona ait kodlar
        Fonksiyona ait kodlar
        Fonksiyona ait kodlar
```

Örnek1: Selamla isminde parametresiz fonksiyon tanımlandı. Selamla() fonksiyonu her çağırıldığında "Merhaba Hoşgeldiniz" Yazısı görülecektir



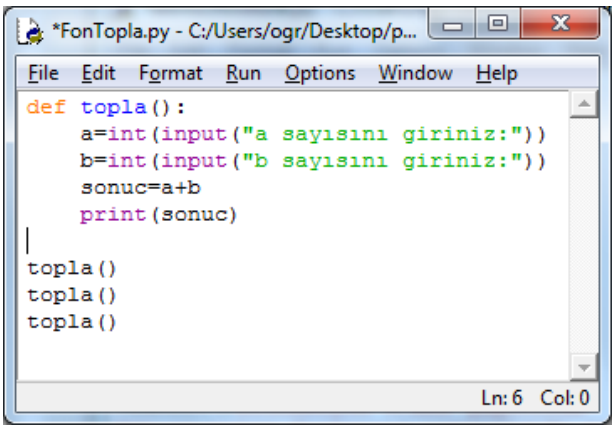

```
Merhaba. Hoşgeldiniz
Merhaba. Hoşgeldiniz
Merhaba. Hoşgeldiniz
>>>
```

ÖRNEK2: Aşağıdaki dosyada 3 adet fonksiyon tanımlandı ve har fonksiyon çağırılarak içindeki kodlar icra edildi

```
#Fonksiyonları tanımlayalım
def menuYaz():
    print("""
    **************
    1-Seçenek 1
    2-Seçenek 2
    3-Seçenek 3
    4-Seçenek 4
    5-Çıkış
    **************
    """)
def OkulAdi():
    print("Anibal Anadolu Lisesi")

def birdenOnaYaz():
    for x in range(1,11):
        print(x)

#Fonksiyonları çağıralım
menuYaz()
OkulAdi()
birdenOnaYaz()
```

```
    **************
    1-Seçenek 1
    2-Seçenek 2
    3-Seçenek 3
    4-Seçenek 4
    5-Çıkış
    **************

Anibal Anadolu Lisesi
1
2
3
4
5
6
7
8
9
10
```

ÖRNEK3: İki sayıyı toplayan parametresiz fonksiyonu oluşturalım. Sayılar Fonksiyon içerisinden input ile alınacaktır. topla() fonksiyonu 3 defa çağırılmıştır.

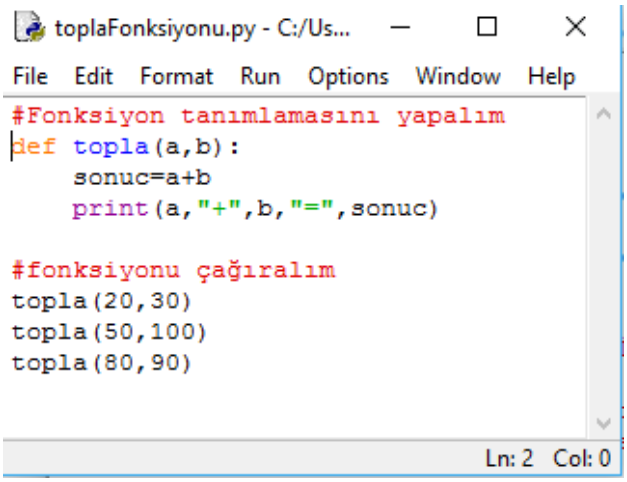



```
a sayısını giriniz:65
b sayısını giriniz:32
97
a sayısını giriniz:23
b sayısını giriniz:21
44
a sayısını giriniz:54
b sayısını giriniz:12
66
>>>
```

PARAMETRELİ FONKSİYONLARIN KULLANILMASI

Parametre alan fonksiyonlar tanımlanırken, Fonksiyon isminin yanındaki parantez içine , kullanılacak parametreler virgülle ayrılarak yazılmalıdır. Parametreler fonksiyon içerisindeki kodlarda kullanılır.

ÖRNEK1: Toplama işlemini parametre alarak yapan fonksiyonun tanımlanması ve çağırılması aşağıdaki gibidir




```
20 + 30 = 50
50 + 100 = 150
80 + 90 = 170
>>>
```

ÖRNEK2: Selamla adındaki fonksiyona gönderilen isim parametresine göre sonuç üretmenktedir.

```
def selamla(isim):
    print("Hoş geldin ",isim)


selamla("Ahmet")
selamla("Mehmet")
selamla("Fatma")
```

```
Hoş geldin  Ahmet
Hoş geldin  Mehmet
Hoş geldin  Fatma
>>>
```

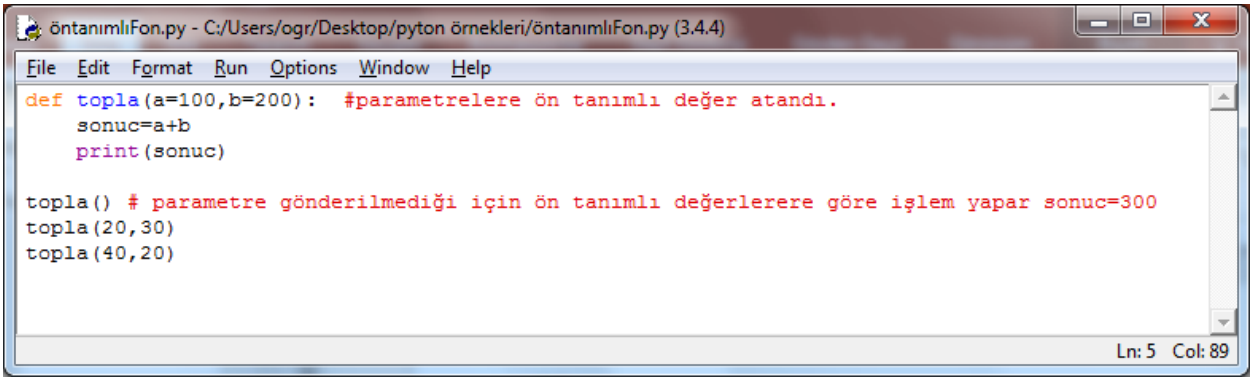
ÖRNEK3: Belirtilen sayılar arasındaki sayıları toplayan parametreli fonksiyon aşağıdaki gibidir

```
def arasiTopla(x,y):
    toplam=0;
    for sayi in range(x,y+1):
        toplam+=sayi
    print(x,"ile", y," arasındaki sayıların toplamı=",toplam)

arasiTopla(1,10)
arasiTopla(1,5)
```

```
1 ile 10  arasındaki sayıların toplamı= 55
1 ile 5  arasındaki sayıların toplamı= 15
```

FONKSİYON OLUŞTURURKEN PARAMETREYE VARSAYILAN DEĞER ATAMA

Fonksiyon içerisindeki parametrelere ön tanımlı değer atanabilir. Aşağıdaki Örneği İnceleyiniz



RETURN İLE FONKSİYONDAN DEĞER DÖNDÜRME

Fonksiyon içerisinde işlenip, üretilen değerler, programın herhangi bir yerinde kullanılmak istenirse «return» komutu ile gönderilmelidir.

```
def kok(sayi):
    sonuc=sayi**0.5
    return sonuc

def usal(sayi,us):
    sonuc=sayi**us
    return sonuc


#denklem için gerekli kodları yazalım

c=kok(usal(3,2)+usal(4,2))

print("bulunan değer=",c)
```

Yandaki örnekte fonksiyon içerisinde üretilen değerler return ile fonksiyonun çağırıldığı yere geri döndürülmüştür

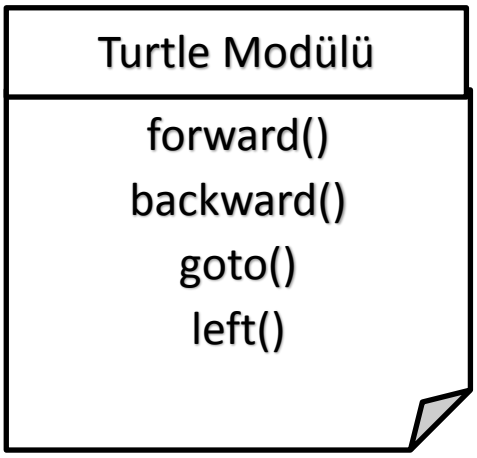
MODÜLLER

Modüllerin, bazı işlevleri kolaylıkla yerine getirmemizi sağlayan birtakım fonksiyonları ve nitelikleri içinde barındıran araçlar olduğunu söyleyebiliriz. Modülleri python dosyasına import ederek içerisindeki fonksiyonları kullanabiliriz.

İki çeşit modül vardır.

1-Hazır Modüller

2-Kendi tanımladığımız modüller

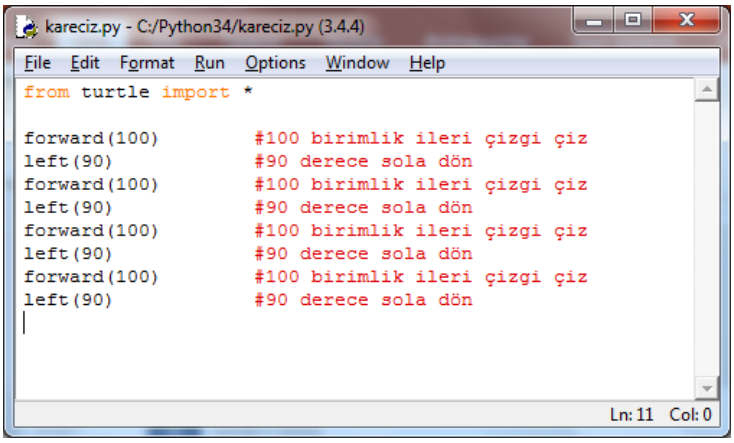


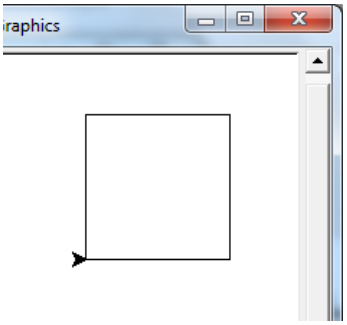
TURTLE MODÜLÜ

Turtle modülünü Python sayfalarımıza import edebilir ve içerisinde yer alan çizim fonksiyonları ile çizimler yapabiliriz. Bu modülü import etmek için en önce aşağıdaki kod satırı yazılmalıdır.

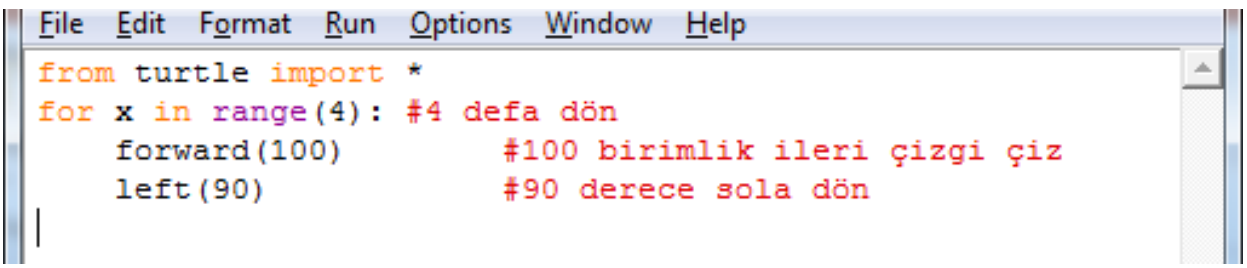
```
from turtle import *
```

ÖRNEK1:Turtle modülünü kullanarak 100 birimlik kare çizelim





Örnek2 : Turtle modülünü kullanarak kareyi for döngüsü ile çizen program



ÖRNEK3: Turtle modülünü kullanarak kareyi fonksiyon oluşturarak ile çizen program(aşağıdaki program aynı yere 3 defa kare çizer. Fonksiyon oluşturulmuş ve 3 defa çağırılmıştır)

kareciz.py - C:/Python34/kareciz.py (3.4.4)

File Edit Format Run Options Window Help

```
from turtle import *          #turtle modülü import et
def kareciz():                #kareciz parametresiz fonksiyonunu tanımla
    for x in range(4):        #4 defa dön
        forward(100)          #100 birimlik ileri çizgi çiz
        left(90)              #90 derece sola dön

kareciz()       #kareciz fonksiyonunu çağır
kareciz()       #kareciz fonksiyonunu çağır
kareciz()       #kareciz fonksiyonunu çağır
```

ÖRNEK4:Girilen kenar sayısına göre 100 ve X birimlik ilgili şekli çizen programı yazalım

sekilciz.py - C:/Python34/sekilciz.py (3.4.4)

File Edit Format Run Options Window Help

```
from turtle import *         #turtle modülü import et
def sekilciz(kenarsayisi,birim):  #şekilciz parametreli fonksiyonunu t
    disAci=360/kenarsayisi  #dönme açısını hesapla
    for x in range(kenarsayisi):    #kenar sayısı kadar  dön
        forward(birim)         #100birim kadar ileri çizgi çiz
        right(disAci)          #hesaplanan derece kadar sağa dön

sekilciz(3,50)  #50 birimlik Üçgen çiz
clear()         #Pencereyi temizler
sekilciz(4,100) #100 birimlik Kare çiz
clear()         #Pencereyi temizler
sekilciz(5,150) #150 birimlik beşgen çiz
clear()         #Pencereyi temizler
sekilciz(6,100) #100 birimlik 6 gen çizer
clear()         #Pencereyi temizler
sekilciz(7,100) #100 birimlik 7gen çizer
clear()         #Pencereyi temizler
```

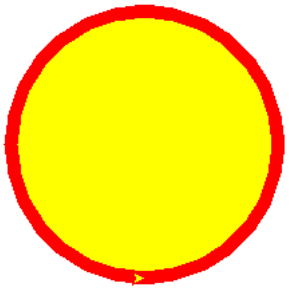
Ln: 16 Col: 30

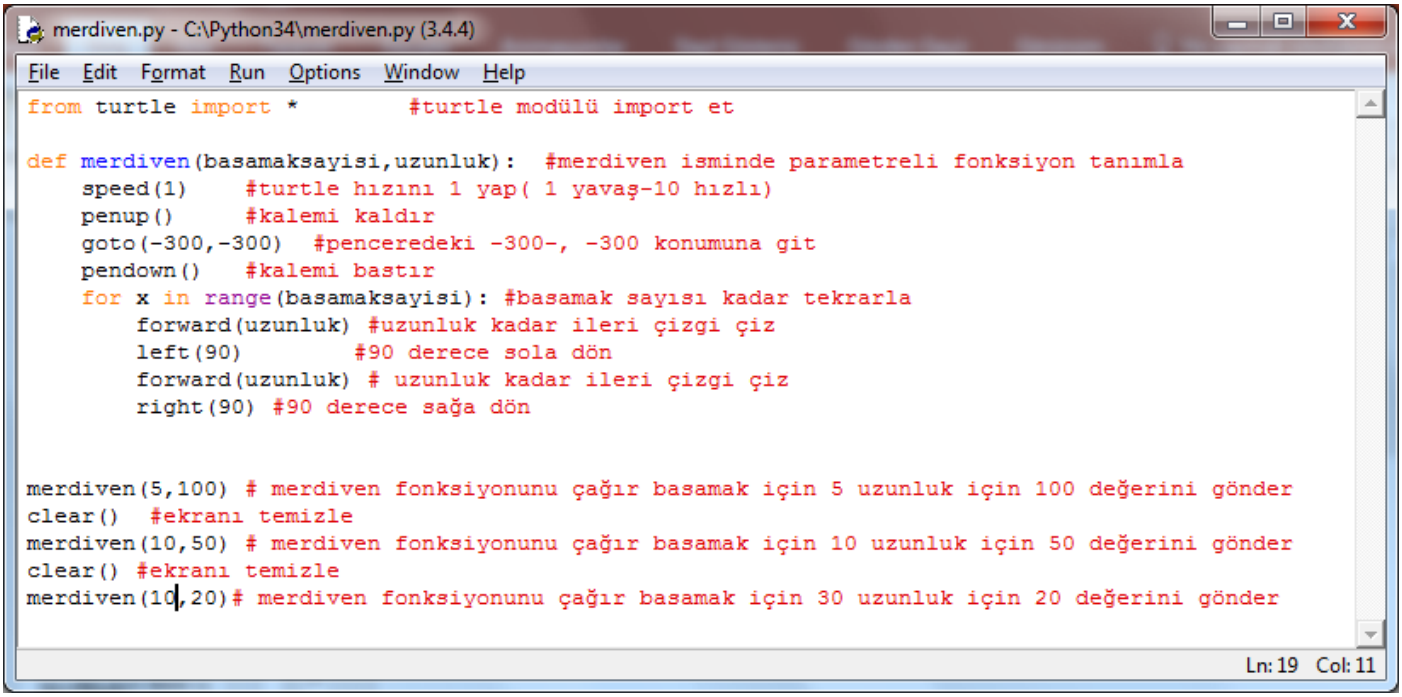
dairerenklendir.py - C:/Python34/dairerenklendir.py (3.4.4)

File Edit Format Run Options Window Help

```
from turtle import *   #turtle modülünü import et

pensize(10)             #kalem kalınlığını 10 birim yap
color("red","yellow")   #color(cizgirenk,dolgurenk) renk tanımla
begin_fill()          #renklendirme başla
circle(100)           #100 birimlik daire çiz

end_fill()            #renklendirme bitir
```

Ln: 3 Col: 24





TUTLE MODÜLÜNDE ÖĞRENDİĞİMİZ FONKSİYONLAR	
Fonksiyon adı-kullanımı	**Açıklaması**
forward(100)	100 birim ileri çizgi çiz
backward(50)	50 birim geri çizgi çiz
left(60)	60 derece sola dön
right(90)	90 derece sağa dön
pensize(10)	Kalem ucu kalınlığını 10 birim yap
color("red","yellow")	Çizgi rengini kırmızı, dolgu rengini sarı yap
begin_fill()	Boyamayı başlat
end_fill()	Boyamayı bitir
circle(50)	50 birimlik daire çiz
speed(1)	turtle hızını ayarla(1 yavaş-10 hızlı)
penup()	kalemi kaldır
pendown()	kalemi bastır
goto(100,200)	pencere de x =100 ,y =200 koordinatına git
clear()	ekranı temizle

https://docs.python.org/3.3/library/turtle.html?highlight=turtle Bu link ile turtle modülü hakkında daha fazla bilgiye ulaşabilirsiniz.

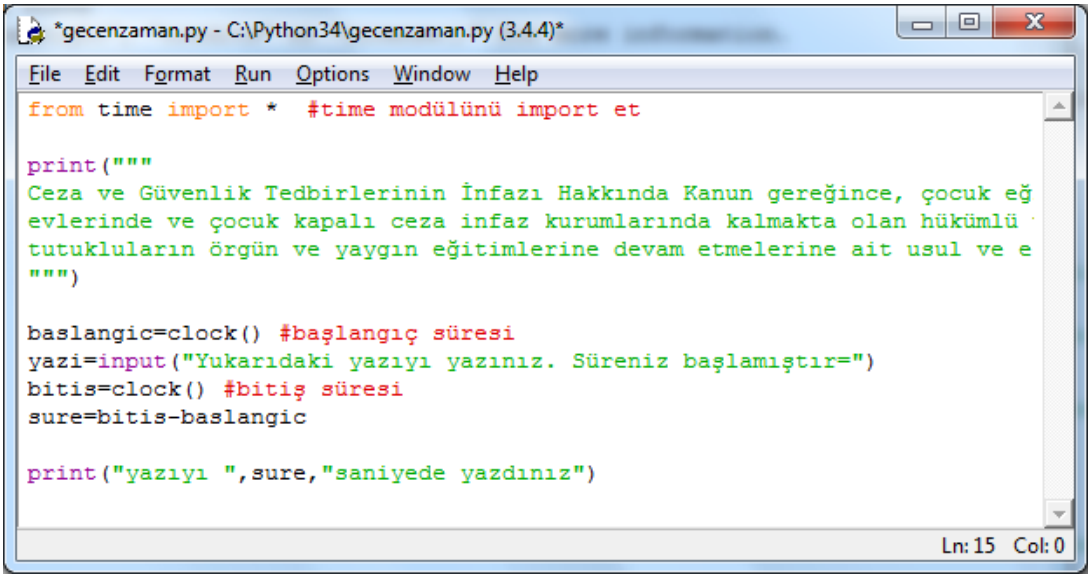
TİME MODÜLÜ

Zamanla ilgili bilgi ve işlemlerin yer aldığı modül, "time" modülüdür. Time modülünü import etmek için aşağıdaki kod satırı yazılmalıdır

```
from time import *
```

clock() fonksiyonu

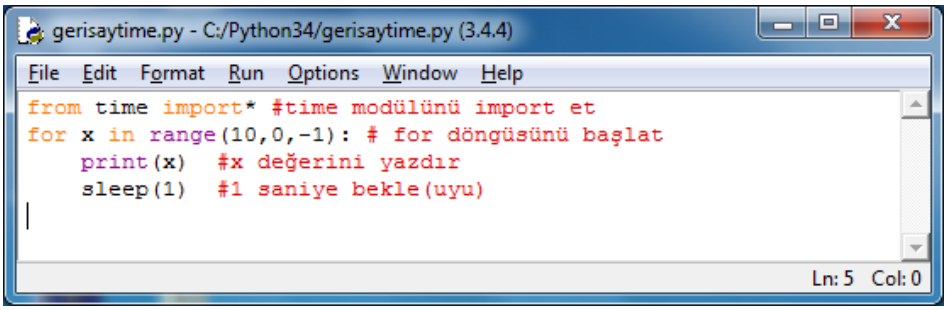
clock() fonksiyonu ile programın belli bölümlerinin çalışma suresini saniye cinsinden ölçebiliriz .iki clock() fonksiyonu arasında geçen süre sn cinsinden hesaplanmaktadır.

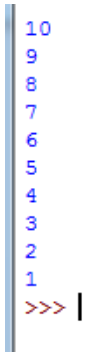


sleep() Fonksiyonu

Sleep() fonksiyonu ise programın çalışması sırasında belirtilen sure kadar durmasını sağlar.

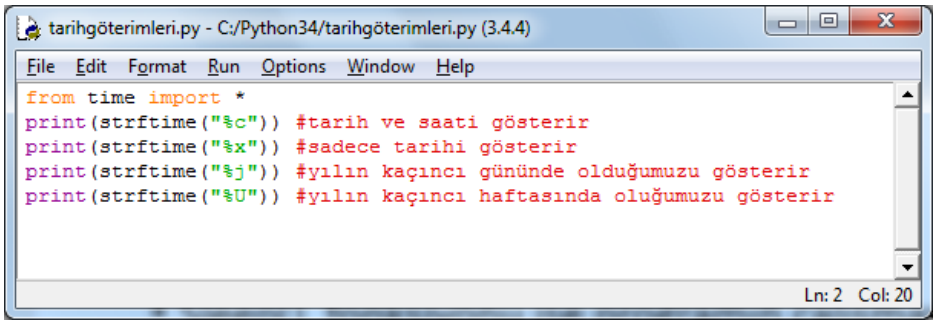
Örneğin geriye sayımda her sayıdan sonra 1 saniye beklemek için aşağıda görülen kod kullanılır.

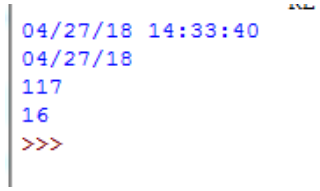




strftime() fonksiyonu

Bu fonksiyon içerisine yazılan özel parametrelere göre zaman hakkında bilgi vermektedir.





RANDOM MODÜLÜ

Bu modülü import ederek bilgisayarın rastgele sayılar üretmesini sağlayabiliriz.Random modülünü import etmek için aşağıdaki satır yazılmalıdır.

```
from random import *
```

```
*randomFonksiyonları.py - C:/Users/ogr/Desktop/pyton örnekleri/randomFonksiyonları.py (3.4.4)*
File  Edit  Format  Run  Options  Window  Help
from random import*
print(random())  #0 ile 1 arasında ondalıklı sayı üretir

print(randint(1,10)) #1 ile 10 arasında ( 1 ve 10 dahil) arasında tamsayı üretir

print(uniform(0.5,2.5)) #belirtilen ondalıklı sayılar arasında sayı üretir

Ln: 4  Col:
```

```
0.5865020954771984
3
1.173638135720657
>>>
```

ÖRNEK1: Rastegele 1 ile 100 arasında 10 adet tam sayı üreten kodları yazalım.

```
rastgele10sayi.py - C:/Python34/rastgele10sayi....
File  Edit  Format  Run  Options  Window  Help
from random import*

for i in range(10):
    print(randint(1,100))

Ln: 4  Col: 25
```

```
52
90
63
56
40
90
97
93
13
24
>>>
```

ÖRNEK2: 1-10 sayıları arasında bilgisayarın üreteceği sayıyı bulmaya çalışalım. For döngüsü ile 3 hakkımız olduğunu unutmayalım.

```
sayı tahminrandom.py - C:\Python34\sayı tahminrandom.py (3.4.4)
File  Edit  Format  Run  Options  Window  Help
from random import*
sayi=randint(1,10)
for i in range(3):
    tahmin=int(input("1-10 arası bir sayı tahmin ediniz:"))
    if tahmin==sayi:
        print("tebrikler bildiniz")
        break
    else:
        print("bilemediniz")


print("bilgisayarın üretiiği sayı=",sayi)

Ln: 1  Col: 0
```

```
1-10 arası bir sayı tahmin ediniz:6
bilemediniz
1-10 arası bir sayı tahmin ediniz:3
tebrikler bildiniz
bilgisayarın üretiiği sayı= 3
>>>
```

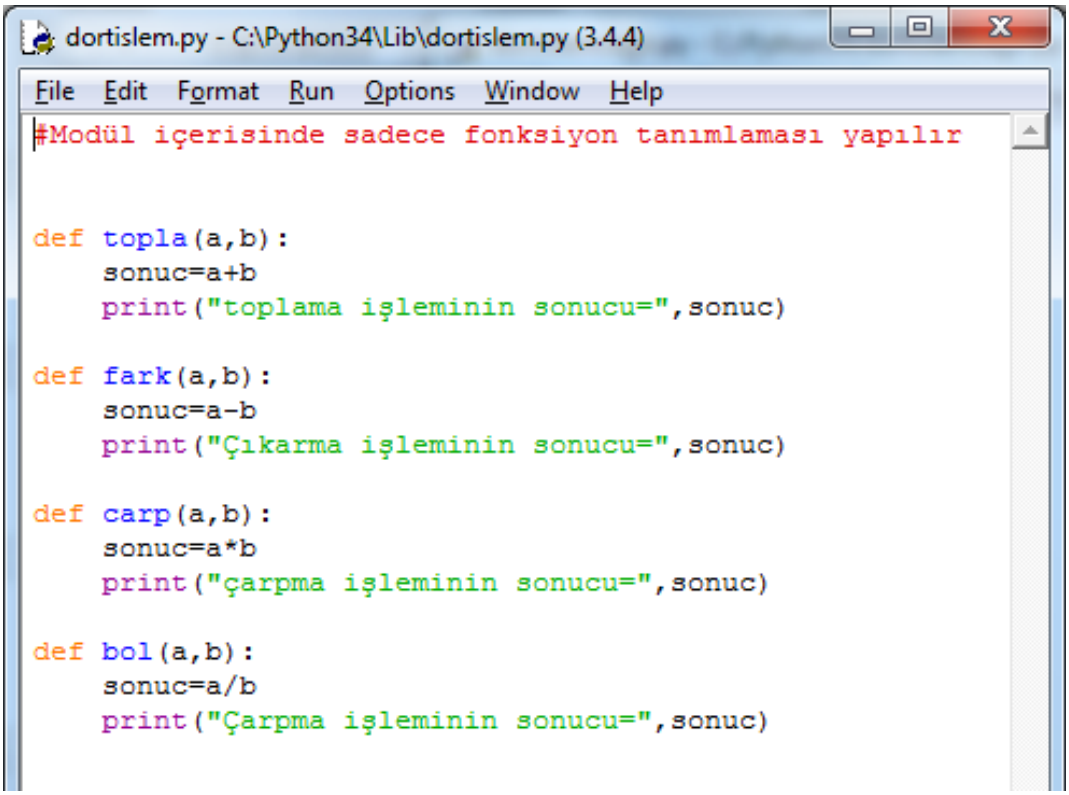
KENDİ MODÜLÜMÜZÜ OLUŞTURALIM

Kendi oluşturduğumuz fonksiyonları modül olarak kaydedebilir ve sayfalarımıza import edebiliriz.

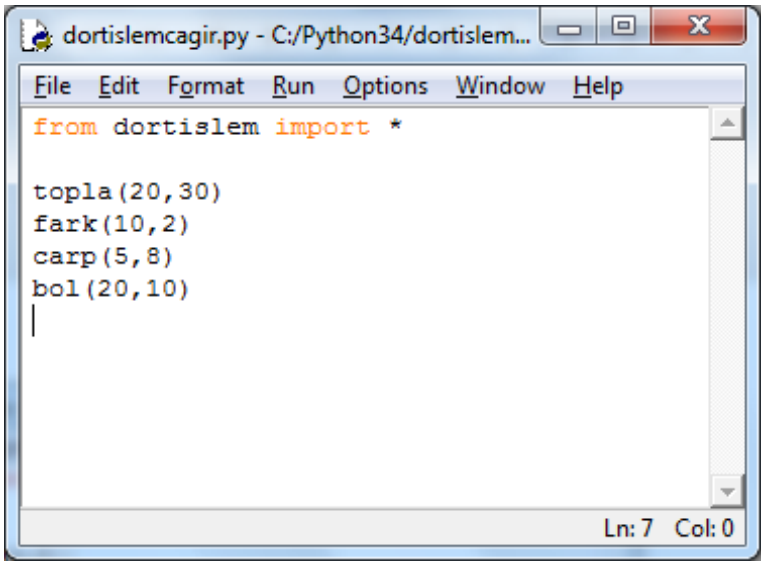
- Bunun için modül içerisinde sadece fonksiyon tanımlaması yapılmalıdır.
- Modül Python>lib klasörüne kaydedilerek tüm çalışmalara import edilebilir.

UGULAMA:

“dortislem" adındaki modülümüz ve içerisinde yer alan fonksiyonlar oluşturuldu ve Bu dosya "lib " klasörüne kaydedildi. Aşağıda “dortislem” adındaki modül dosyamızı görüyorsunuz.



Daha sonra modül import edilerek içerisindeki fonksiyonlar çağırılıp kullanılır. Aşağıda bunun ile ilgili kodları görebilirsiniz




```
toplama işleminin sonucu= 50
Çıkarma işleminin sonucu= 8
çarpma işleminin sonucu= 40
Çarpma işleminin sonucu= 2.0
>>>
```