

# 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 TIPLERİ

Şimdiye kadar 3 veri tipi gördük

VERİ TİPİ	AÇIKLAMA	ÖRNEK
String veri tipi	Metin , yazı ve karakter dizileri string veri tipine örnektir	"Ahmet" "Anibal" "54kljlişliş87" "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(karakterSayisi)

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 operatörler

Matematiksel işlemleri yaparken kullanılan operatörler 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

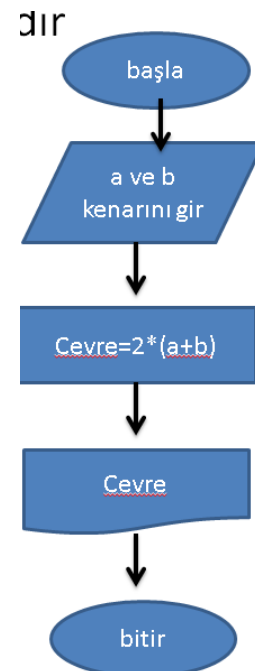
- 1.adım:başla
- 2.adım:a ve b kenarlarının değerini gir
- 3.adım:çevreyi hesapla>> çevre=2\*(a+b)
- 4.adım:çevreyi göster-yazdır
- 5.adım:bitir

## İş Akış Şeması

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

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

Şekil 1.10: Akış şeması sembolleri



## 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

```
karsilastirmaopereatorleri.py - C:\Users\ogr\Desktop...
File Edit Format Run Options Window Help
a=int(input("a sayısını giriniz:")) #a değişkeni
b=int(input("b sayısını giriniz:")) #b değişkeni

print("a küçüktür b-----",a<b)
print("a büyüktür b-----",a>b)
print("a eşittir b-----",a==b)
print("a küçük veya eşittir b-----",a<=b)
print("a büyük veya eşittir b-----",a>=b)
print("a eşit değil b-----",a!=b)

Ln: 10 Col: 0
```

```
*Python 3.6.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\ogr\Desktop\ornekler\karsilastirmaopereatorleri.py ====
a sayısını giriniz:6
b sayısını giriniz:3
a küçüktür b----- False
a büyüktür b----- True
a eşittir b----- False
a küçük veya eşittir b----- False
a büyük veya eşittir b----- True
a eşit değil b----- True

Ln: 15 Col: 0
```

## 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

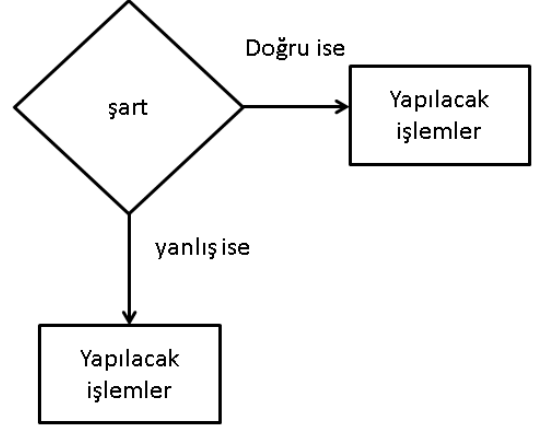
Koşul yanlış(false) ise else altındaki satırlar çalışır

### 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

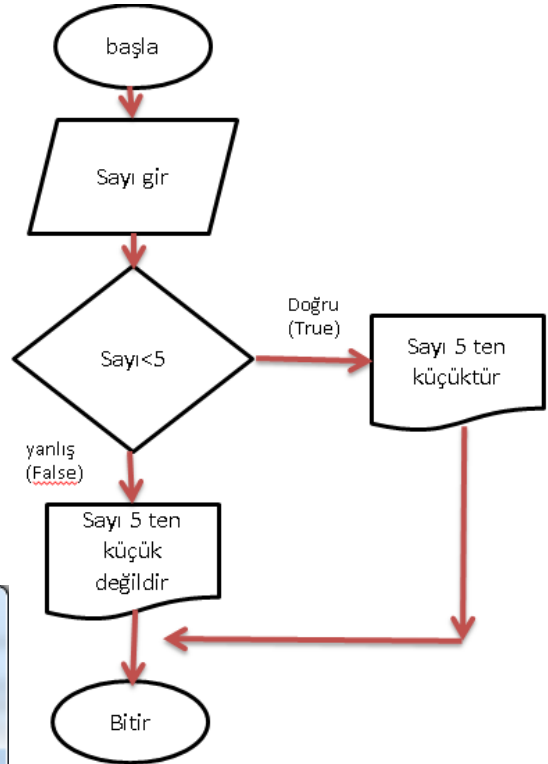
1.adım: Başla

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



```
*if5tenkucukmu.py - C:\Users\ogr\Desktop\ornekler\if5tenkucukmu.py
File Edit Format Run Options Window Help
sayi=int(input("Lütfen sayı giriniz"))

if sayi<5:
    print("girilen sayı 5 ten küçüktür")
else:
    print("girilen sayı 5 ten küçük değildir")
|
```

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ogr\Desktop\ornekler\if5tenkucukmu.py =====
Lütfen sayı giriniz:9
girilen sayı 5 ten küçük değildir
>>> |
```

## 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

```
elifkullanimi.py - C:\Users\ogr\Desktop\ornekler\elifkullanimi.py (3.6.2)
File Edit Format Run Options Window Help
sayi=int(input("1 ile 5 aralığında bir sayıs giriniz"))

if sayi==1: # ilk olarak her zaman if ile başlıyoruz eğer say
    print("bir")
elif sayi==2: # eğer sayı 2 eşitse
    print("iki")
elif sayi==3: # eğer sayı 3 eşitse
    print("üç")
elif sayi==4: # eğer sayı 4 e eşitse
    print("dört")
elif sayi==5: # eğer sayı 5 e eşitse
    print("beş")
else: # hiç biri değilse
    print("girdiğiniz sayı 1 ve 5 aralığında değil")

Ln:1 Col:0
```

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017,
it (Intel)] on win32
Type "copyright", "credits" or "license()"
>>>
===== RESTART: C:\Users\ogr\Desktop\or
=====
1 ile 5 aralığında bir sayıs giriniz:1
bir
>>>
===== RESTART: C:\Users\ogr\Desktop\or
=====
1 ile 5 aralığında bir sayıs giriniz:2
iki
>>>
===== RESTART: C:\Users\ogr\Desktop\or
=====
1 ile 5 aralığında bir sayıs giriniz:3
üç
>>>
===== RESTART: C:\Users\ogr\Desktop\or
=====
1 ile 5 aralığında bir sayıs giriniz:9
girdiğiniz sayı 1 ve 5 aralığında değil
>>> |
```

## 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 sisteme giriş yapılacaktır. değilse, «kullanıcı adınız veya parolanız yanlış» çıktısını veren kodları yazınız.

```
ifandkullanimi.py - C:\Users\ogr\Desktop\ornekler\ifandkullanimi.py (3.4.4)
File Edit Format Run Options Window Help
kAdi=input("Kullanıcı adını giriniz:")
parola=input("Parolanızı giriniz:")

if kAdi=="admin" and parola=="abcdefg":
    print("Sisteme hoşgeldiniz")
else:
    print("Kullanıcı adı veya parolayı hatalı girdiniz")
```

## «And» işlecinin mantıksal gösterimi

if şart1 and şart2:

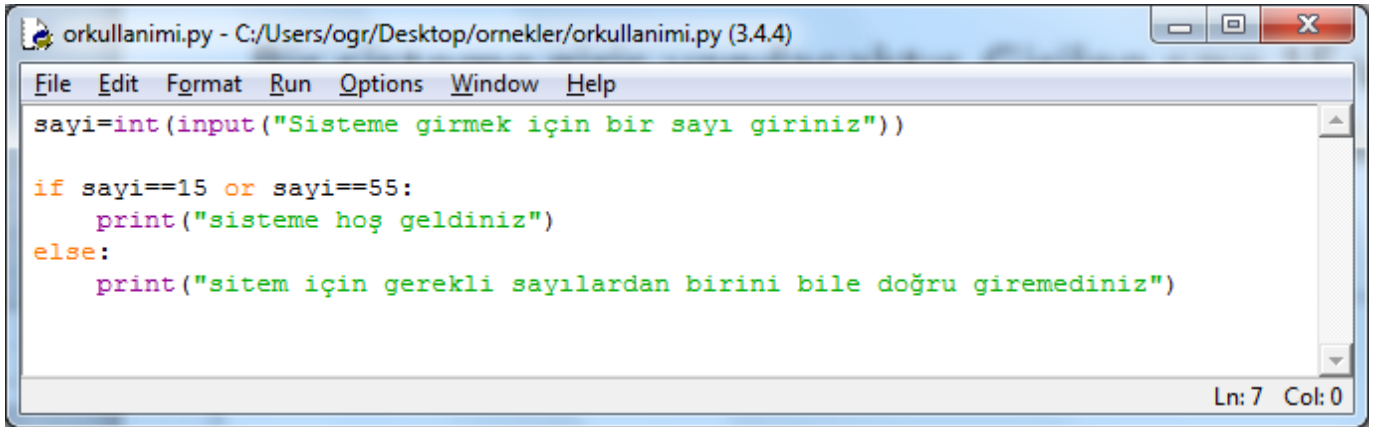
Şart1	şart2	sonuç	İşletilecek yer
True	True	True	İf altındaki satırlar
True	<u>False</u>	<u>False</u>	Else nin altındaki satırlar
<u>False</u>	True	<u>False</u>	Else nin altındaki satırlar
<u>False</u>	<u>False</u>	<u>false</u>	Else nin altındaki satırlar

## 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



```
orkullanimi.py - C:/Users/ogr/Desktop/ornekler/orkullanimi.py (3.4.4)
File Edit Format Run Options Window Help
sayi=int(input("Sisteme girmek için bir sayı giriniz"))
if sayi==15 or sayi==55:
    print("sisteme hoş geldiniz")
else:
    print("sitem için gerekli sayılardan birini bile doğru giremediniz")
Ln: 7 Col: 0
```

## «or» işlecinin mantıksal gösterimi

İf şart1 or şart2:

Şart1	şart2	sonuç	işletilecek yer
True	True	True	<u>İf</u> altındaki satırlar
True	<u>False</u>	True	<u>İf</u> altındaki satırlar
<u>False</u>	True	<u>true</u>	<u>İf</u> altındaki satırlar
<u>False</u>	<u>False</u>	<u>false</u>	Else <u>nin</u> 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.

```
print(10<20)
print(not 10<20)
```

```
===== RESTART: C:\Users\ogr\Desktop\ornekler\
notiledeğilinalma.py =====
True
False
>>>
```

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

```
puan=float(input("Sınav Puanınızı giriniz"))
if not puan<50:
    print("Geçti")
else:
    print("kaldı")
|
```

## not (değil ) işlecinin mantıksal göstesi

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.

```
File Edit Format Run Options Window Help
print("h" in "ahmet")
print("s" in "ahmet")
```

```
===== RESTART: C:/Users/ogr/Desktop
/ornekler/inTest.py =====
True
False
>>> |
```



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

```
email=input("Lütfen email adresinizi 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="auıoüeiö"  
harf=input("Bir karakter giriniz")  
if harf in sesliHarfler:  
    print("Girdiğiniz karakter sesli harftir")  
else:  
    print("Girdiğiniz 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 ...  
#     Girdiğiniz 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.

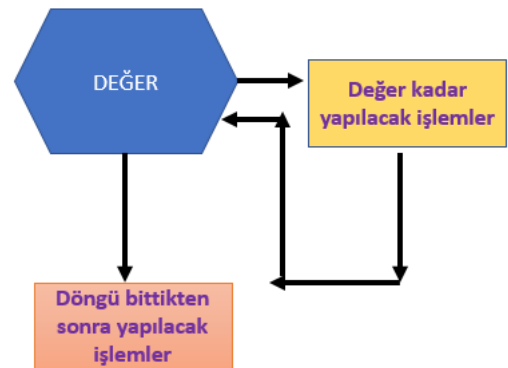
For döngüleri belirtilen sayıdaki işlemlerin tekrarlanması için kullanılan döngülerdir. For döngüleri başlangıç ve bitiş değerleri arasında artım miktarına göre istenilen sayıda tekrar yapar.

**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
<pre>for sayi in range(5):     print(sayi)</pre>	0,1,2,3,4	<b>range(değer)</b> 0 dan başlar, değer 1 eksiğine kadar sayı üretir. Bu döngü 5 kez- defa döner
<pre>for i in range(1,11):     print(i)</pre>	1,2,3,4,5,6,7,8,9,10	<b>range(başlangıç, bitiş)</b> başlangıç değeri ile başlar, bitiş değerinin 1 eksiğine kadar döngü devam eder.1 den 10 'a kadar sayıları yazar. 11 dahil değildir.
<pre>for sayi in range(5,20,3):     print(sayi)</pre>	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	<pre>for i in range(10):     print("anibal")  </pre>
girilen ismi , girilen sayı kadar (defa) yazdıran program	<pre>isim=input(" isim giriniz") kez=int(input("kaç defa yazılsın"))  for x in range(kez):     print(isim) </pre>
50 den 100'e kadar sayıları yazdıralım	<pre>for sayi in range(50,101):     print(sayi)  </pre>
0,100 e kadar 5 er artarak yazdıran programın kodlarını yazalım	<pre>for sayi in range(0,100,5):     print(sayi)</pre>

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

```

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

```

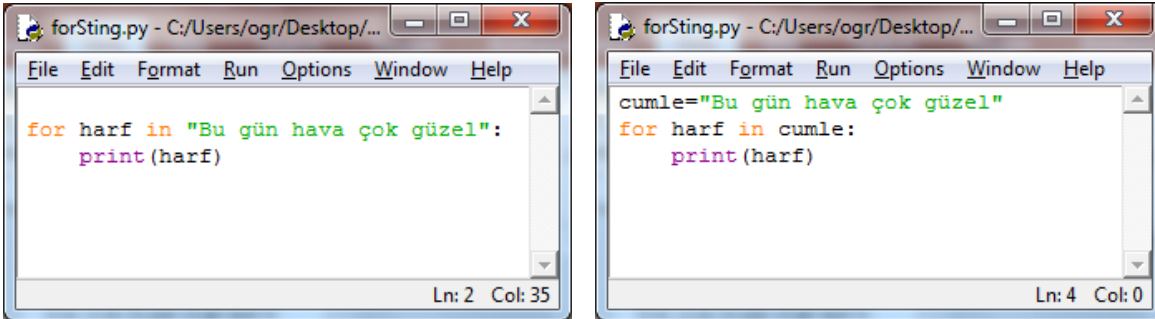
toplam=0 # döngüden gelen sayılar toplam değişkenin üzerine eklenecek
for sayi in range(1,101):
    toplam+=sayi # bu satırda sayılar toplam değişkeninin üzerine ekleniyor
print("Toplama işleminin sonucu:",toplam)

```

## FOR DÖNGÜSÜNDE STRING VERİLERİN KULLANILMASI

For döngüsü ile string veri içerisindeki her bir karakteri alabiliriz

Aşağıdaki örnek ile belirtilen string ifade döngü içerisine adını ve her harf tek tek alınarak yazdırıldı



Örnek: Girilen cümle içerisindeki sesli harflerin sayısını bulalım

```

sesliHarfler="auoüeiö"
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 atılıyor
#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

```

H A Z İ R A N 2 0 2 2  
 K O Ç  
 >>>

## 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):          1
    print(sayi)                  2
    if sayi==5:                  3
        break                    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 atadıktan 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 içerisinde 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