

## Input () Fonksiyonu:

Kullanıcıdan değer almak için kullanılır. Alınan değerler string tipindedir.

### Örnek:

```
a=input("birinci sayıyı giriniz: ")
b=input("ikinci sayıyı giriniz: ")
print("girdiğiniz sayıların toplamı: ",a+b)
birinci sayıyı giriniz:
6 ikinci sayıyı giriniz:
girdiğiniz sayıların toplamı: 68
```

Örnekte görüldüğü üzere uygulama bize hatalı bir çıktı vermiştir. Bu kullanımda her iki değer bir karakter dizisi olarak algılanmış ve Python iki değer üzerinde toplama işlemi yapamadığı için yan yana yazarak birleştirmiştir.

### Örnek:

```
a=int(input("birinci sayıyı giriniz: "))
b=int(input("ikinci sayıyı giriniz: "))
print("girdiğiniz sayıların toplamı:",a+b)
```

```
birinci sayıyı giriniz: 5
ikinci sayıyı giriniz: 8
girdiğiniz sayıların toplamı: 13
```

Görüldüğü üzere burada yapılan işlemle girdiler sayılara dönüştürülmüş ve toplama işlemi yapılmıştır. input( ) fonksiyonu kullanılırken sık yapılan hataların başında hatalı veri girişleri gelmektedir. int( ) fonksiyonu ile beraber integer ifade yerine string bir değer girilirse Python hata verir. Çünkü metinsel ifadeler (a, b, isim, soyisim) sayılara dönüştürülemez.

### Örnek:

```
a=int(input("birinci sayıyı giriniz: "))
b=int(input("ikinci sayıyı giriniz: ")) print("girdiğiniz
sayıların toplamı:",a+b)
```

```
birinci sayıyı giriniz: 8
ikinci sayıyı giriniz: Ali
```

Traceback (most recent call last):

```
File "C:/Users/asus/Desktop/1.py", line 2, in <module> b=int(input("ikinci sayıyı
giriniz: "))
```

ValueError: invalid literal for int() with base 10: 'Ali'

## if yapısı

Bu yapıda, belirli komutların çalışması, bir koşula bağlıdır. Koşul sağlanmazsa herhangi bir işlem yapılmaz.

Kullanımı: Aşağıda bir "if" bloğu gösterilmektedir "if" bloğunun dikey hizasının sağında olan kod satırları koşul gerçekleştiğinde çalışır. Bu kodlar "if" bloğunda yer almaktadır. Büyük eşittir operatörü karşılaştırma sonucu "boolean" (True veya False) bir değer verir. True değer verirse "if" bloğu içinde (girintide olan) kodlar çalışır. Koşul sağlanmazsa yani "False" değeri verirse bloğun içine girilmez bloktaki kodlar atlanır.

Örnek:Kullanıcının yaş değerini alarak 18'e eşit veya büyük olması hâlinde ona mesaj veren kod:

```
yasi=int(input('Lütfen yaşınızı giriniz: '))
```

```
if (yasi>=18):
```

```
    print('Oy kullanabilirsiniz.')
```

```
print ('Program bitti.')
```

```
Lütfen yaşınızı giriniz: 18 Oy
```

```
kullanabilirsiniz.
```

```
Program bitti.
```

"if" bloğunun içindeki kod ancak şart sağlandığında çalışır ve blok bittikten sonra program akışı devam eder. Şart sağlanmazsa blok atlanır. Python alt satırdaki kodları yorumlar ve ona göre işlem yapar.

```
yasi=int(input('Lütfen yaşınızı giriniz: '))
if (yasi>=18):
    print('Oy kullanabilirsiniz.')
```

```
print ('Program bitti.')
```

```
Lütfen yaşınızı giriniz: 15
Program bitti.
```

Örnek: "adi" ve "yasi" değişkenlerinin değerlerine göre hangi blokların çalıştığına dikkat ediniz.

```
yasi=17
adi='Tahsin'
if (yasi>=18):
    print('1. Şart sağlandı.')
```

```
    print ('1. if bloğunun içindesiniz.')
```

```
if (adi=='Tahsin'):
    print('2. Şart sağlandı')
```

```
    print ('2. if bloğunun içindesiniz.')
```

```
print('Normal program akışı girinti yok.')
```

```
2. Şart sağlandı.
```

```
2. if bloğunun içindesiniz.
```

```
Normal program akışı girinti yok.
```

Koşul ifadelerinde birden fazla koşul birlikte kullanılabilir. Bunun için operatörler konusunda açıklanan mantıksal operatörler kullanılır. Örnekte “and” operatörü ile iki koşulun birlikte sağlanma şartı koşulmuştur

Örnek : Kullanıcının girdiği kullanıcı adı ve parolayı kontrol ediniz.

```
kullaniciAdi=input('Kullanıcı Adı:')
kullaniciParola=input('Parola:')
if (kullaniciAdi=='Admin' and kullaniciParola=='123456'):
    print('Giriş başarılı.')
    print ('Menülere erişebilirsiniz.')
Kullanıcı Adı:Admin Parola:123456
Giriş başarılı.

Menülere erişebilirsiniz.
```

\*\*\*\*\*Örnekte şartlardan biri bile sağlanmazsa “if” bloğundaki kodlar çalışmaz.\*\*\*\*\*

## if-else Yapısı

“if” yapısında şart sağlanırsa blok içindeki kodlar çalışmaktadır. Ancak şartın sağlanmadığı durumlarda herhangi bir işlem yapılmaz. “else” ifadesi değilse anlamındadır. Yani şartın sağlanmadığı durumda çalışacak kodlar “else” bloğuna yazılır.

Kullanımı: “else” bloğu da if bloğu gibi ayrı bir blok olarak yazılır. Bir “if” bloğundan sonra gelen else bloğu aynı girinti seviyesinde olmalıdır. “else” bloğu “if” ile birlikte kullanılır.

Aşağıdaki örnekte kullanıcının girdiği sayının çift - tek olduğunu bulan bir kod yazılmıştır. Çift bir sayı girildiğinde “if” bloğunun içindeki kodlar çalışır.

```
sayil=int (input ('Lütfen bir sayı giriniz: '))
if ((sayil%2)==0):
    print('Girdiğiniz sayı
çifttir.') else:
    print('Girdiğiniz sayı tektir: ')
Lütfen bir sayı giriniz: 12
Girdiğiniz sayı çifttir.
```

Tek bir sayı girildiğinde “else” bloğunun içindeki kodlar çalışır.

```
sayil=int (input ('Lütfen bir sayı giriniz: '))
if ((sayil%2)==0):
    print('Girdiğiniz sayı çifttir.')
else:
    print('Girdiğiniz sayı tektir: ')
Lütfen bir sayı giriniz: 13
Girdiğiniz sayı tektir:
```

### Örnek:

Koşullu ifade, operatörler ve bağlaçlarla daha etkili yapılabilir. Yukarıdaki kullanıcı adı ve parolası örneğinde kullanıcının girdiği kullanıcı adı ve parola bu örnekte “and” ile birlikte kontrol edilmiştir.

```
kullaniciAdi=input('Kullanıcı Adı:')
kullaniciParola=input('Parola:')
if (kullaniciAdi=='Admin' and kullaniciParola=='123456'):
    print('Giriş başarılı.')
    print ('Menülere erişebilirsiniz.')
else:
    print ('Yanlış kullanıcı adı veya şifre')
Kullanıcı Adı:Admin
Parola:123456
Giriş başarılı.
Menülere erişebilirsiniz.
```

Örnek: “if else” yapısına ilişkin başka bir örnek aşağıda verilmiştir. Bu örnekte şartlar “and” bağlacıyla birleştirilmiştir.

```
yasi=int(input('Lütfen yaşınızı giriniz: '))
bolum='Bilgisayar'
yabanciDil=True
if (yasi>=18 and yasi<35 and bolum=='Bilgisayar' and yabanciDil==True):
    #Aşağıdaki kodun çalışması için yukarıdaki 3 şartın da sağlanması
    gerekir.
    print('Mülakata katılabılırsınız.')
else:
    print('Şartlarınız tutmuyor.')
Mülakata katılabılırsınız.
```

## if-elif-else Yapısı

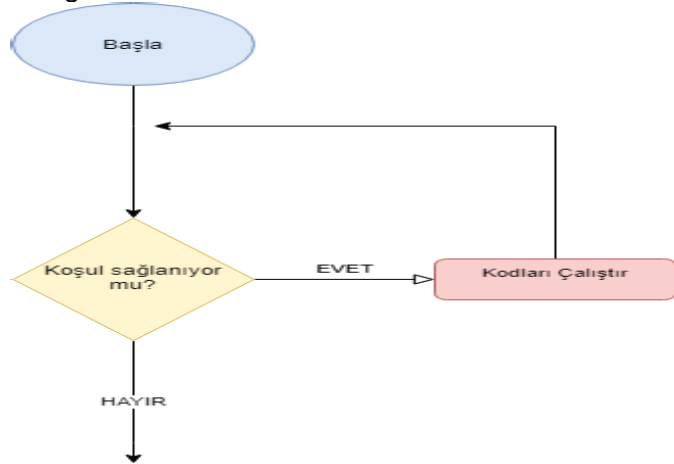
Bu yapıda koşullar art arda verilir. if ile verilen koşulun devamında ‘değilse şu ise’ anlamına gelen “elif” ifadesi yer alır. Yapının en sonunda ise ‘hiçbiri değilse’ anlamında else ifadesi yer almaktadır. Her ifade kendi bloğundaki kodları çalıştırır. “if, elif ve else” bloklarının girinti düzeyleri aynı olmalıdır.

Her koşul ifadesi bir “if” bloğu formatında yazılabilir ancak bu durumda program akışında tüm koşul ifadeleri tek tek kontrol edilirdi. “if-elif-else” yapısında ise şart sağlandığında veya else ifadesine gelindiğinde ilgili bloktaki kodlar çalışır ve tüm “if-elif-else” bloğundan çıkarılır. Birbirleriyle bağlantısı olmayan koşullar ayrı “if” blokları şeklinde verilebilir. Ama bir değer belirli aralıktaki şartları sağlayıp sağlamadığı kontrol edilirken “if-elif-else” yapısını kullanmak daha uygundur. Bu yapıda koşullardan biri sağlanıyorsa diğer koşullar kontrol edilmez. Alınan değer “if-elif-else” yapısındaki yalnız bir koşulu sağlayabilir.

Örnek: Bir kullanıcının sınav puanını alarak durumunun değerlendirilmesi

```
sinavPuanı=int(input('Puanınız giriniz (0-100): '))
if sinavPuanı>=85:
    print('Pek iyi')
elif sinavPuanı>=70:
    print('İyi')
elif sinavPuanı>=55:
    print('Orta')
elif sinavPuanı>=45:
    print('geçer')
else: print('Kaldı')
Puanınız giriniz (0-100): 65
Orta
```

### Döngüler- while



While döngüsü, koşul gerçekleştiği sürece çalışan bir döngü çeşididir. Genellikle döngünün kaç defa çalışacağı belirli değilse while döngüsü tercih edilir. Ancak koşullar verilerek de while döngüsünün belirli sayıda çalışması sağlanabilir. Döngülerde koşullu ifadelerde olduğu gibi blok yapısı kullanılmaktadır. while ifadesinden sonra koşul durumu yazılır, ardından iki nokta işareti konularak alt satıra geçilir. Koşul durumu sağlandığı sürece çalışacak kodlar bir blok içeriden çalışır.

while(koşul durumu):

1. adım
2. adım
3. adım
- .
- .

Bu durum bir örnekle incelenmek istenirse,

### Örnek 1:

```
# şartın başlangıç değeri
sayac=1

#sayac 6 dan küçük olduğu sürece
while sayac<6:

    print("merhaba dünya")
    sayac=sayac+1
```

```
merhaba dünya
merhaba dünya
merhaba dünya
merhaba dünya
merhaba dünya
```

Örnek 1’de sayac isimli değişkenin değeri 1’den başlamış, yine değişken değeri 6’dan küçük olduğu sürece konsol üzerinde “merhaba dünya” yazılır. Sayacın değeri 1 arttırılarak döngünün başına döner ve sayac değeri 6’ya eşit olana kadar bu durum devam eder. Aynı işlemi ekrana sayac isimli değişkenin değeri yazılarak yapmak istenirse,

### Örnek 2:

```
sayac=1
while sayac<6:
    print(sayac)
    sayac=sayac+
1
2
3
4
5
```

## Döngünün Kapsamı

Hatırlanacağı üzere döngü koşulunun sağlandığı sürece daha içteki bloklarda bulunan kodların çalışacağı belirtilmişti. Döngü bittiği zaman Python bir üstteki bloğa dönerek çalışmasına devam eder.

### Örnek 3:

```
# şartın başlangıç değeri
sayac=1
#sayac 6 dan küçük olduğu sürece
while sayac<6:
    print(sayac)
    sayac=sayac+1
#döngü bittiği zaman
print("döngü sonlandı")
1
2
3
4
5
döngü sonlandı
```

while döngüsü ile çalışırken sık yapılan hatalardan birisi döngü içerisinde koşulu sağlayan değişkenin değerini arttırma işleminin unutulmasıdır. Bu durumda koşul sürekli sağlanacağı için döngü sürekli çalışır ve dışarıdan bir müdahale ile sonlandırılması gerekir.

Örnek 4 :

```
sayac=1
while sayac<6:
    print(sayac)
```

Yukarıdaki örnekte bilinçli olarak sayaç isimli değişkenin değeri arttırılmamıştır. Uygulamayı çalıştırıldığında görüleceği üzere program hiç durmadan çalışacak ve ekrana sürekli 1 değeri yazacaktır.

Örnek5: while döngüsü kullanarak 1-100 arasındaki (100 dâhil) çift sayıları bularak ekrana yan yana yazan programı yazınız.

```
a=1
while a<=100:
    if a%2==0:
        print(a,end=","
) a=a+1
```

2, 4, 6, 8... , 100

Örnek 6: while döngüsü kullanarak 1 – 100 arasındaki sayıların toplamını bulan programı yazınız.

```
toplam=0
i=1
while i<=100:
    toplam=toplam+i
    i=i+1
print("sayıların toplamı",toplam)
sayıların toplamı 5050
```

Örnek 6'da yapılan işlemde toplam isimli bir değişken oluşturularak başlangıç değeri 0 olarak belirlenmiştir. Çünkü toplam değerini hesaplarken toplam=toplam+i şeklinde bir işlem yapılmaktadır. Eğer toplam tanımlı olmasaydı, tanımlanmamış\_değer=tanımlanmamış\_değer+sayı şeklinde bir işlem yapılmaya çalışacak ve hata verecekti. Yani bir değişken ile işlem yapılmadan önce tanımlanması gerekmektedir.

## While True – Break İfadeleri ve Sonsuz Döngüler

Program yazarken bazen döngünün ne zaman sonlanacağı bilinmeyebilir. Örnek olarak bir markette müşterilerin alışveriş yaparak sepetlerini doldurdukları ve sepette kaç adet ürün olduğu bilinmeyebilir. O müşteriye ait tüm ürünler barkod okuyucu ile okutulmalı ve

toplam tutar hesaplanmalıdır. İşte bu gibi belirsiz durumlar için while döngüsü ile beraber True ifadesi ya da benzer yapılar kullanılabilir. Break ifadesi ise döngü sürekli çalışırken istenilen bir anda döngüden çıkmak için kullanılır.

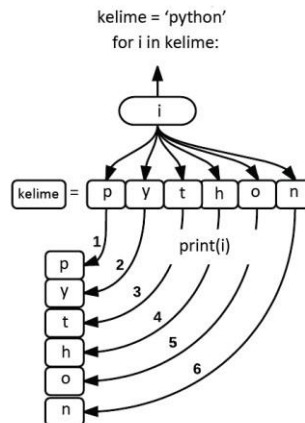
```
i=1
while True:
    print(i)
    i+=1
    if i==6:
        break
print("döngü sonlandı")
1
2
3
4
5
döngü sonlandı
```

Örnek: Girilen sayının faktöriyelini hesaplayan programı yazınız.

```
i=1
f=int(input("faktöriyeli alınacak sayıyı giriniz: "))
sonuc=1
while i<=f:
    sonuc=sonuc*i
    i+=1
print(sonuc)
faktöriyeli alınacak sayıyı giriniz: 5
120
```

## For Döngüsü

For döngüsü, Python'da genellikle döngünün tekrar sayısı programcı tarafından belirlenmiş veya öngörülmüş ve belli ise kullanılır. Hatırlanacağı üzere while döngüsü ile sonsuz döngüler yapılabiliyor ve istenilen bir anda döngüden çıkılabiliyordu. For döngüsü daha çok belirli sayıdaki işlemi gerçekleştirmek için kullanılır. Bunun yanında Python'da for döngüsünün iterasyon denilen önemli bir özelliği bulunmaktadır. İterasyon işlemi sayesinde karakter dizileri ve listeler üzerinde gezinme işlemi, yani ilk elemandan son elemana kadar işlem yapabilmektedir. For döngüsü kullanmak için "in" işlecinden faydalanmak gerekmektedir.





Örnek:

```
isim="Mustafa"
for i in isim:
    print(i,end=",")
M,u,s,t,a,f,a,
```

## range Fonksiyonu ile For Döngüsü Kullanımı

Python'da for döngüsüyle belirli değerler arasında döngü kurmak istenirse range fonksiyonu kullanılmalıdır.

```
for i in range(2,12,2):# range fonksiyonu ilk değer, son değer, artış miktarı
şeklindedir.(isa). Son değer her zaman bir alt değerde biter.
    print(i)
çıktı:2,4,6,8,10
```

```
for i in range(3,12):# range fonksiyonunda iki değer varsa bunlar ilk değer ve
son değerdir.artış miktarı ise 1 dir.
    print(i)
çıktı:3,4,5,6,7,8,10,11
```

```
for i in range(12): # range fonksiyonunda tek değer varsa bu değer son
değerdir.ilik değer sıfırdır ve artış miktarı 1 dir.
    print(i)
Çıktı:0,1,2,3,4,5,6,7,8,9,10,11
```

```
for i in range(10,0,-1): # 20 den geriye doğru sayar
    print(i)
Çıktı:10,9,8,7,6,5,4,3,2,1
```

Fonksiyonlar:

- 1- def kullanılmak zorunda
  - 2- fonksiyon ismi olmak zorunda
  - 3- parantez aç kapa olmak zorunda
  - 4- : olmak zorunda
  - 5- fonksiyona parametre girmek zorunlu değildir. Tanımlanan parametre sayısı kadar parametre ile fonksiyon çağırılır.
- ```
def fonksiyonİsmi (parametre1,paarametre2):
    pass
```

## Örnek Sorular:

### Doğru yanlış soruları:

- 1- input fonksiyonu string tipinde değer döndürür.
- 2- `a=int(a)` ile a değişkenini integer tipinde bir değişkene çeviririz
- 3- while ile sonsuz döngüler oluşturabiliriz
- 4- break komutu ile döngü sonlandırılır
- 5- continue komutu ile o anki işlem biter ancak döngü bir sonraki işlemde devam eder.
- 6- and ile yapılan işlemlerde iki şartında doğru olması gerekir.
- 7- or ile yapılan işlemlerde şartın birinin doğru olması yeterlidir.
- 8- if ile istenen şart sağlanmazsa blok içindeki işlemler yapılmaz.
- 9- `a % b` ile a'nın b'ye bölümündeki kalanı buluruz.
- 10- elif demek else-if demektir.
- 11- for döngüsü range fonksiyonunda son değer olmak zorundadır.
- 12 – for döngüsü range fonksiyonu en fazla 3 değer alır.
- 13- for döngüsü range fonksiyonu ilk değer, son değer ve artış miktarı şeklindedir.
- 14- for döngüsü range fonksiyonunda 2 değer varsa bunlar ilk değer ve son değerdir.
- 15- for döngüsü range fonksiyonunda tek değer varsa bu son değerdir.
- 16-for döngüsü range fonksiyonunda değer girilmezse artış miktarı 1, ilk değer sıfırdır.
- 17-listelerde indis değeri 0 ile başlar
- 18- listelerde integer ve string değerleri birlikte tutabiliriz.
- 19- listeye eleman eklemek için append kullanırız.
- 20-listede sıralama yapmak için sort kullanırız.
- 21-fonksiyon oluşturmak için def anahtar kelimesini kullanırız.
- 22- fonksiyonun çalışması için çağırmak gerekir.
- 23-her fonksiyonda parametre olmak zorunda değildir.
- 24- fonksiyona değer girmek için kullanılan değişkenlere parametre denir.
- 25-kaç parametre ile fonksiyon oluşturduysak aynı sayıda parametre ile çağırmalıyız

### Örnek çoktan seçmeli sorular:

1- Hangisi python programlama dilinde değişkenin değerini bir artırır?

- a- `a+=1`                      b- `a++`  
c- `a=+1`                      d- `a=a++`

2- `a=39` , `b=a//10` `c=a%10` `toplam= a+b+c` ise toplam nedir?

- a- 50                              b- 51  
c- 52                              d-53

3- `for i in range (2,9)` sonucu nedir?

- a- 3,4,5,6,7,8                  b- 2,3,4,5,6,7,8  
c- 3,4,5,6,7,8,9                d- 2,3,4,5,6,7,8,9

4- `a=1`  
`while True:`

`print (a)`                      programında çıktı ne olur?

- a- 1 yazar                      b- a yazar  
c- çıktı üretmez                d- sonsuz defa 1 yazar

5-`for i in range (10, 3, -2)` sonucu nedir?

- a-10,8,6,4,2,0,-2              b- 10,8,6,4,2  
c- 10,8,6,4                      d- 10,8,6,4,2,

6-sayac=0  
while sayac<6:  
  sayac+=1  
  if sayac==3 and sayac==4 :  
    continue  
  print(sayac)

**Programının çıktısı nedir?**

a-1,2,3,4,5,6      b-0,1,2,3,4,5,6  
c- 1,2,5,6          d-0,1,2,3,4,5

7-Aşağıdaki programın çıktısı nedir?  
sayac=0

```
toplam=0
while sayac<5:
  sayac+=1
  toplam=toplam+sayac
print(toplam)
```

a-8      b-12      c-15      d-20

8- liste=[4,4,3,6,8]

toplam=0

for i in liste:

  toplam+=i

ortalama=toplam/len(liste)

print(ortalama)

programının çıktısı nedir ?

9-def toplama(a,b):

  Print(a+b)

işleminin sonucu nedir ?

10- a="m e r h a b a"

for i in a:

  if i!=" ":

    print(i,end="")

programı nasıl bir çıktı üretir ?

Başarılar dilerim.  
Serkan GÜNER  
Bilişim Teknolojileri Öğretmeni