DÖNGÜLER (LOOP): **Birden fazla kod dizisinin tekrar tekrar çalışmasını sağlayan yapılardır. 2 farklı döngü çeşidi vardır. Bunlar for ve while döngüsüdür.**

for döngüsü: Verileri veya nesneleri tek tek işleme sokar. Örnek:

#1'den 5' kadar olan sayıları yazdırmak isteyelim.

```
print(1)
print(2)
print(3)
print(4)
print(5)
```

Çıktı:

1
2
3
4
5

#Amacıma ulaştım. Ancak, 1'den 100'e kadar yazmak gerekseydi böyle bir çözüm yolu doğru olmayacaktı. Bu durumda döngü yapıları tercih edilmelidir.

#Şimdi de 1'den 100'e kadar olan tüm sayıları yazdıralım. Değişkenimiz i olsun;

```
for i in range(1,101):
    print(i, end=" ")
```

Çıktı: 1 2 3 . . . . . 97 98 99 100

Ek bilgi: Buradaki i değişkeni sembloliktir. Aslında istenilen isimlendirme yapılabilir. Ancak genel alışkanlık olarak i harfi kullanılır. Örneğin n harfi de sıklıkla kullanılır. Bu değişken döngüye girecek olan tüm veri veya nesneleri temsil eder.

Ek bilgi: **end=" "** → end parametresi verilerin sonuna neyin ekleneceğini belirler. Burada sadece boşluk var. Böylece her veri işlendikten sonra sonuna boşluk eklenecektir.

Ek bilgi: Döngülerde ve koşullu(if) yapılarda iki altın kural vardır. Döngünün başladığı satırın sonuna iki nokta üst üste konulur. Satır aşağısında yazılan kodlar ise biraz içeriden yazılır.

range fonksiyonu: Türkçe olarak "aralık" anlamına gelmektedir. Python'da sayı aralıklarını belirtmek için kullanırız. Temel olarak çalışma mantığı şu şekildedir:

range (başlangıç değeri, bitiş değeri, atlama sayısı)

Birkaç tane örnek verelim:

range(4,10,2): Burada başlangıç değeri 4, bitiş değeri 10, atlama sayısı ise 2'dir. Burada 4'den başlayarak bitiş değerine kadar 2'şer 2'şer sıra atlanır. Ancak bitiş değeri olan 10 bu

gruba dâhil değildir. Çünkü range fonksiyonunda bitiş değeri, belirtilmişse işleme alınmaz.
Sonuç: 4-6-8

range(1,10): Burada başlangıç değeri 1, bitiş değeri 10'dur. Ancak atlama sayısı belirtilmemiştir. Burada 1 ile 10 aralığındaki sayılar kastedilir. Ancak buna 10 dâhil değildir. Çünkü range fonksiyonunda bitiş değeri belirtilmişse işleme alınmaz.
Sonuç: 1-2-3-4-5-6-7-8-9

range(10): Burada başlangıç değeri ve atlama sayısı belirtilmemiştir, bitiş değeri ise 10'dur. Burada 0 ile 10 aralığındaki sayılar kastedilir. Çünkü range fonksiyonunda başlangıç değeri yoksa sıralama 0'dan başlar. Bitiş değeri belirtildiği için yine bu gruba 10 dâhil edilmez.
Sonuç: 0-1-2-3-4-5-6-7-8-9

range(10,2,-2): Burada başlangıç değeri 10, bitiş değeri 2, atlama sayısı ise -2'dir. Bunun anlamı başlangıç değerinden itibaren 2'şer 2'şer azalma olmasıdır.
Sonuç: 10-8-6-4

#Şimdi de 1'den 100'e kadar tüm tek sayıları yazdıralım. Değişkenimiz bu sefer n olsun;

```
for n in range(1,101,2):
    print(n, end=" ")
```

Çıktı: 1 3 5 . . . . . 95 97 99

Açıklama: range(1, 101, 2) ifadesindeki 1 başlangıç sayısıdır. Eğer burası boş bırakılırsa sayı otomatik olarak sıfırdan başlar. 101 ise yazılacak sayıların sınırıdır. 101 sayısı çıktıya dâhil değildir. 2 ise artış miktarını gösterir. Yani sayıyı 2'şer arttırır.

Ör:

```
for n in range(21,0,-3):
    print(n, end=" ")
```

Çıktı: 21 18 15 12 9 6 3

#1'den 100'e kadar olan sayıların toplamı

```
top= 0
for i in range(1,101):
     top+= i
print(top)
```

Çıktı: 5050

#Değişken ve for kullanımı:

```
sayılar = "12345"
for sayı in sayılar:
      print(int(sayı) * 2, end=" ")
```

Çıktı: 2 4 6 8 10

#Değişken ve if kullanımı:

```
sayilar = "1234567"
for i in sayilar:
    if int(i) > 3:
        print(i, end=" ")
```

Çıktı: 4 5 6 7

Açıklaması: sayilar değişkeni oluşturduk. "1234567" ifadesindeki her bir karakteri ayırdık. Yani tüm sayılar artık bağımsızlığını ilan etmiş durumda. Ayrıca her bir karakteri i değişkene atadık. Ancak biz bu karakterleri int(i) kodunu yazarak tamsayıya çevirdik. Çünkü matematiksel işlemler yapmak için verileri sayıya çevirmemiz gerekiyor. if komutuyla da 3'ten büyük olan sayıları yazdırdık.

#for ve liste kullanımı:

```
a=[]
for i in range(2,10,2):
    a+=[i]
print(a)
```

Çıktı: [2, 4, 6, 8]

#for ve liste kullanımı ile dataların sadece ilk harflerini alalım.

```
isim=["Onur", "Duru", "Teoman"]
for i in isim:
    print(i[0])
```

Çıktı: O D T

while döngüsü: Bir koşul sağlanmaya devam ettiği sürece işlemleri tekrarlar. İngilizce bir kelime olan while, Türkçede '-iken, olduğu sürece' gibi anlamlarına gelir.

Örnek kullanım:

```
while a == 1:
```

Anlamı: a, 1'e eşit olduğu olduğu sürece (şu işi yap)

#sonsuz döngü (infinite loop)

```
a = 1
while a < 5:                    #a, 5'dan küçük olduğu sürece ekrana "Duru" yazdır.
        print("Duru")
```

Açıklaması: Burada a, 5'den küçük olduğu sürece ekrana "Ali" yazdıracak. Ancak sorun şu ki a=1 olduğu için a her zaman 5'den küçük olacak. Bu da ekrana sürekli "Ali" yazılmasına neden olacak. Buna sonsuz döngü diyoruz. (infinite loop) Buna son vermek için klavyenizde Ctrl+C veya Ctrl+Z tuşlarına basarak programı durmaya zorlayabilirsiniz.

#1'den 5'e kadar olan sayıları yazdıralım.

```
a=1
while a<5:                  #a 5'ten küçük olduğu sürece
    print(a, end=" ")       #a'yı yazdır.
    a+=1                    #a'yı 1 arttır.
```

Çıktı: 1 2 3 4

Açıklaması: İlk satırda a değişkene 1 değerini atadık. İkinci satıra geldiğimizde değişkenin 5'ten küçük olup olmadığına baktık, eğer küçükse kodumuz alt satıra geçecek ve böylece değişken ekrana yazdırılacak. Son koda geldiğimizde a değişkeni 1 değer artıp 2 olacak ve döngüye girecek. Sonra döngü devam edip ekrana 2 yazdırılacak. Bu durum a'nın 5'ten küçük olmaması şartına kadar sağlanacak. Yani a artık 5 olduğunda döngü duracak.

#Döngünün kaç defa döneceğini kullanıcın girdiği sayı ile belirleyelim.

```
n = 1
karar= int(input("Sayılar kaça kadar sıralansın? " ))
while n <= karar:
      print(n)
      n += 1
```

Kullanıcı etkileşimi: Sayılar kaça kadar sıralansın? Cevap: 7

Çıktı: 1 2 3 4 5 6 7

Açıklaması: Bu programda döngünün kaç defa döneceğine kullanıcı karar verecektir. Örneğin kullanıcı 7 sayısını girerse 1'den 7'ye kadar olan sayılar ekrana yazdırılacaktır. Ancak kullanıcının girdiği sayı 1 veya daha az olursa program tepki vermeyecektir.

while True: Bu kodu kullandığımızda döngü aksi belirtilmediği müddetçe sonsuza kadar çalışır.

Örnek:

```
while True:
    print("Bilgisayar çıldırdı!")          #ekrana sonsuza kadar "Bilgisayar çıldırdı!" yazısı gelir.
```

Çıktı: Sonsuz döngü…

#Ancak break komutu ile sonsuz döngüye son verebiliriz.

```
while True:
    print("Bilgisayar çıldırdı!")
    break
```

Çıktı: Bilgisayar çıldırdı!

#Aşağıdaki toplama işlemi sürekli tekrar edilecektir.

```
while True:
    a=int(input("sayı gir: "))
    b=int(input("sayı gir: "))
    print("Sayıların toplamı:", a+b)
```

#Kullanıcının önceden belirli bir veri girişi yapana kadar programın sürekli çalışması

```
while True:
    cevap = input("Nasılsın, iyi misiniz? : ")
    if cevap == "Tamam":
        print("Tamam kızma sormuyorum artık! ")
        break
```

#Kullanıcıya bir parola belirletirken parolanın 6 karakterden kısa olmamasını sağlayalım. Ayrıca parola girilmezse uyarı versin.

```
while True:
    parola = input("Bir parola belirleyin: ")
    if not parola:
        print("Parola bölümü boş geçilemez!")
    elif len(parola) < 6:
        print("Parola 6 karakterden kısa olmamalı")
    else:
        print("Yeni parolanız: ", parola)
        break
```

continue komutu: Türkçe olarak “devam” anlamına gelen bu kelime görüldüğü zaman diğer kodları es geçilip döngünün başına gelinir. Örnek:

#Sınıf listesinden Metehan adlı kişinin çıkartılması gerekiyor.

```
sinif=["Onur", "Serkan", "Metehan", "Ezgi"]
for i in sinif:
    if i=="Metehan":    #eğer döngüde “Metehan” varsa onu atla ve diğerlerine devam et
            continue
    print(i, end=" ")
```

Çıktı: Onur Serkan Ezgi

Örnek:

```
a=[1,2,3,4,5]
for i in a:
    if i==3:
        continue
    print(i, end="-")
```

Çıktı: 1-2-4-5-

pass komutu: Herhangi bir işlem yapmadan geçeceğimiz durumlarda kullanılır. Kısaca "Hiçbir şey yapmadan yola devam et!" anlamı taşır. O an için karar veremediğimiz ama daha sonra karar verdiğimiz işlemleri de pass komutu yerine yazabiliriz.

break komutu: Programı tamamen sonlandırır.

#sayı tahmin oyunu

```
sayi=["9"]
while True:
    tahmin=input("1'den 10'a kadar bir sayı tahmin et: ")
    if tahmin in sayi:
        print("Tebrikler doğru tahmin!")
        print("Oyun bitti!")
        break
    else:
        pass
```