**Loops (Döngüler) :**

Döngüler tekrar eden işlemleri gerçekleştirmek için tasarlanmış yapılardır. Bu nedenle döngü yapıları, farklı kaynaklarda tekrarlı yapılar olarak da adlandırılmaktadır. Programcı, yazdığı programın bazı kod satırlarını tekrarlı olarak çalıştırma ihtiyacı duyduğunda döngü yapılarını kullanır.

**while döngüleri** : tekrar sayısının koşula bağlı olduğu yani koşul gerçekleştiği veya gerçekleşmediği sürece işlemlerin tekrar ettiği döngülerdir. Tekrar sayısı baştan bilinmediğinden, burada tekrar sayısı koşula bağlı olarak gerçekleşir.

**for döngüleri :** terkarlama işlemin kaç kez yapılacağı baştan belli olduğu dögülerdir.

**while Döngüsü :** işlemlerin ne kadar tekrarlanacağı baştan belli olmadığı ancak bir koşulun doğru ya da yanlış olmasına bağlı olarak tekrar sayısının belirlendiği döngülerdir.

while koşul: <koşul>:geçerli olduğu sürece tekrarla

Blok işlemleri Verilen koşul doğru olduğu sürece döngü bloğu içerisindeki işlemler tekrarlanır. Koşul kontrol işlemi, döngü başında yapılır. Koşul yanlış olduğunda döngü sonlanır.

*Döngü değişkenini başlatma (i=10)*

*while(mantık cümlesi):*

*(girinti) Yapışıcak işlemler*

*Artırma İşlemleri*

**while True :** Sonsuz döngü oluşturur. Döngü içine koşulsuz giriş sağlanırken döngüden çıkış ancak break komutu ile gerçekleşir. Döngü içerisnde kod satırları hep aynı hizada (4 karakterlik (tab)) aynı girintiye sahip olmalıdır. Bir kod bloğu içerisinde baş kod bloklarıda yer alabilir fakat her yeni kod bloğu 4 karakter daha sağa hizalanmalıdır. while döngülerinde döngüden çıkışı gerçekleştiricek koşulu sağlayan bir ifade döngü içerisinde bulunmalıdır. Aksi takdirde program çıktı vermezsensiz döngüye girer (while True gibi), sonsuz döngülerden de çıkış,döngü kırılarak break komutu ile gerçekleşir.

**break ve continue Komutları** : break ve continue komutları döngülerde ek kontrol sağlar. break komutu; bir koşula bağlı olarak döngü değişkeninin son değerine ulaşmadan döngüyü terk etmesini sağlar. continue komutu ise döngü içerisinde o anki işlemi atlayarak göz ardı etmemizi sağlar fakat döngüden bir çıkış sağlamaz.

**break Komutu** : Bir koşula bağlı olarak while ve for döngülerden çıkmak için yani döngüleri sonlandırmak için kullanılır. Program akışını döngünün düışındaki ilk kımuta atlatır.

**continue Komutu** : contunie komutu; döngü içerisinde o anki işlemin atlanması ve bir sonraki işlemden devam edilmesini sağlar.

**for Döngüsü :** Tekrar sayısının baştan belli olduğu, ardışık eleman listesinin sırayla işleme konulduğu döngü yapısıdır.Liste/dizi gibi sıralı elamanlar üzerinde işlem yapılır. for döngüsün yapısında yer alan "range, in..." gibi komutlar bulunur. Ekran çıktısı alt alta değil yan yana yazılsın isteniyorse " end="" " print komutuna eklenir.

**range() Fonksiyonu** : range(ilk,son,adım) şeklinde kullanılır.

* print(range(10)) ekran çıktısı 0123456789
* range(5) : 0'dan 5'e kadar sayılar. 5 hariç
* range(3,15): 3'den 15'e kadar sayılar. 3 dahil 15 hariç
* range(3,15,2): 3'den 15'e kadar sayılar ikişer artarak işlem gerçekleştirilir. 3 dahil 15 hariç
* reversed(range(3,15,2)): 15'den 3'e sayılar ikişer azalatak işlem gerçekleştirir.

**in/not Operatörleri** : Sıralı yapılarda (range, list, tuple yapıları gibi) eleman sorgulanamsında kullanılır.

* X in S : Eğer X,S içerisinde (S:sıralı bir yapıdır) varsa True, değilse False döndürür.
* X not in S : Eğer X, S içerisinde (S: sıralı bir yapıdır) varsa False, değilse True döndürür.

**index(x) Fonksiyonu :** range() fonksiyonu gibi sıralı bir yapı içerisindeki x elamanın indisini döndürür.

r=range(0,20,2)

print(11 in r) #False

print(10 in r) #True

print(r.index(10)) #5

for x in range (3,26,2):

print(x) Rakamaları alt alta değil de **yan yana** yazdırmak için print(x) komutunun yerine print**(x,end='')** şeklinde kodlanması gerekir.

**enumerate() Fonksiyonu** : for döngüsü sırasında sıralı semboller (sözel ifadeler) kümesinde aynı anda hem sembol değerini hem sırasını (indisini) öğrenmek için kullanılan bir fonksiyondur.

for i, gun in enumerate(["pazar","cumartesi",]) :

print(i,gun)

**İç İçe Döngüler :**

Birden fazla döngü ve seçimli yapı iç içe kullanılabilir. Sırlama, matris veya tablo gibi iki-üç işlemlerin yapıldığı uygulamalrda sıklıkla iç içe döngüler kullanılır. İç içe döngülerin yapıları, bir döngü içersinde birden fazla döngünün kullanıldığı yapılardır. İçteki döngü tamamen bitmeden dıştaki döngü dıştaki döngüye gezilmez.

**Seriler ve Uygulama Örnekleri :**

Matematikte x1+x2+x3+x4+......+xn ifadesine seri denir. Serinin her bir elamanına terim denir. Bir seriyi oluşturan elemanların toplamı işlemi 'yığmalı toplama' işlemidir.

*ST=0*

*n=int(input("Terim Sayısı : "))*

*for x in range(1,n+1):*

*ST=ST+1/x*

*print("Toplam= ",ST)*