**Karakterler Diziler ve String İşlemleri**

"String" kelimesi bazı kaynaklarda Türkçeye "katar,dizge dize"gibi sözcüklerle tercüme edilir. String değişkenler; Unicode karakterlerin (harflerin,sayıların, ve +,-,&, gibi diğer özel simgelerin) saklandığı sözel/metinsel değişkenlerdir. C dilindeki karşılığı ile bir string aynı zamanda bir karakter dizgisidir. Python dilinde değişkenleri str veri tipi ile ifade edilirler. Python dili çift tırnak "" tek tırnak '' veya üç tırnak '''...''' içerisinde yazılan her şeyi string olarak algılar.

**Stirng Biçimlendirme Operatörlerü : %**

String içersinde sayılal veya sözel bir değişken içerğini yazdırmak için % operatörü kullanılabilir. Tam sayı değişkenlerini ifade etmek için %d, string değişkenlerini ifade etmek için %s, kesirli ifadeleri ifade etmek için %f (sadece virgülden sonraki 2 basamağı ifade etmek için %2.f) kullanılır. String ifadeden sonra yazdırlıcak değişken isimlerinin önüne % operatörü konmalıdır.

**Tek Bir Stringin (Karakter Dizgisin) Elemanların Erişim**:

String işlemlerde ve sıralı/ardışık listelerde değişken ismini takiben parantez içersinde belirtilen numarya indis adı verilir. Bir strigin her bir elamanına ayrı dizilerde/listelerde olduğu gibi indis nuamrsı ile erişebiliriz.

**Stringlerin Dilimelme ve Birleştirme:**

Stringler üzerinde toplama, çarpma işlemleri gerçekleştirebiliriz. İki string ifadeyi birleştimek için + ifadesi kullanılır.

**String Dilimleme Operatörü: (:)**

Stringleri parçalara ayırmak için ':' operatörü kllanılır. String dilimeleme işleminde ':' operatörü kullanım şekli. strDizi [ilkİndis: sonİndis: adımMiktarı] Eğer ilk indis değeri belirtilmezse '0', son indis belirtilmezse 'dizinin toplam uzunluğu' kabul edilir. Adım miktarının varsayılan miktarı '1' dir. S[1::2] 1. indisli elemandan başla ve ikişer atlayarak dizi sonuna git. S[1:6:3] 1. indisli elemandan başla ve 6.indisli elemana kadar 3'er 3'er git.

**Bir Stringi Silmek :**

Python, bir stringin elemanlarının doğrudan silinmesine izin vermez. Fakat bir stringi bir bütün olarak silebilirsiniz. Bunun için **del** gibi fonksiyonları kullanabilirsiniz. Bir stringi silemiyoruz ama stringi listeye dönüştürüp, pop fonksiyonu ile indis numarası vererek silebilirsiniz.

**Python String/Sayısal Dönüşümleri :**

Python'da string/sayısal dönüşümlerde; dönüştürülücek ifade o veri tipi sarmalına alınır. Örneğin, integer veri tipini string veri tipine dönüştürürken ifade 'int' sarmalına alınırken; tersi durumunda 'str' sarmalına alınır.

**DÜZENLİ İFADELER (REGULAR EXPRESSİONS-RE MODÜLÜ)**

Bir string veya metin içerisinde; Python regex(düzenli ifadeler) kullanarak bu ifadelere uygun sorgulamalar yapılabilir. Regex özellikle stringlerle uğraşan programcılar için büyü kolaylık sağlar. String parçalama, ayıklama, belli kriterleye uyan verileri alma (e-mail,password kontrolü), arama (anahtar kelime araması), değiştirme gibi işlermlerde regex kullanılabilir. Bü işlemler için 're' modülü içersinde tanımlı bazı özel karakterler ve metotlar kullanılmaktadır.

**re Modülü:**

Python dilinde düzenli ifadeleri kullanabilmek, yönetmek için re modülü kullanılır. re modülünü; import re şeklinde program başında çağırmak gerekir. Tablo(9.2'de re modülüne ait metotlar var. Bu metotların listesini standart metotların isimleri ayrıştıtılarak dir(re) komutu ile elde edilir.)

**Mach** : bir metni başka bir metin içerisinde arar, bulamazsa None değer döndürür. re.match(“ol”,s)

**Fullmatch :** metnin tamamı ile eşleşmezse None değer döndürür. re.fullmatch(“bir ol”,s)

**Split :** Bir metni belirtilen kalıba göre parçalar. Re.split(“ ”,s)

**Sub :** bir string’i başka bir stringle değiştirir. Re.sub(“sarkar”,”kalkar”,tekerleme)

**Findfall** : re.findall(“\w+olur”,siir) # sonu olur ile biten kelimeleri döndürür.