\n parametresi: Bu parametreye newline adı verilir. print() fonksiyonu içerisinde kullanıldığında ilgili yerden bir alt satıra geçiş yapar. Aşağıdaki örneği inceleyelim.

```
print("bilgisayar bilimi")
Çıktı: bilgisayar bilimi
```

Yukarıdaki örnekte **\n** karakterini araya koyarsak ilgili yerden bir alt satıra geçiş yapacaktır. Yukarıdaki örneği bir de şu şekilde yazalım.

Ör:

```
print("bilgisayar\nbilimi")
Çıktı:
bilgisayar
bilimi
```

#Görüldüğü üzere **\n** parametresini bilgisayar ifadesinin hemen sonuna koyduk ve böylece tam da oradan satır başına geçiş yapmış olduk.

\t parametresi: print() fonksiyonu içerisinde kullanıldığında ilgili yerden bir tab kadar boşluk bırakır.

Ör:

```
print("Ocak\tŞubat\tMart")
Çıktı: Ocak     Şubat   Mart
```

*** parametresi:** stringi parçalara böler.

Ör:

```
print(*"Onur")
Çıktı: O n u r    (Yani buradaki her bir karakter O,n,u,r artık bağımsız birer string oldu)
```

sep parametresi: İngilizcede separator (ayırıcı, ayraç) kelimesinin kısaltmasıdır.

```
print("www.", "google.", "com")
Çıktı: www.  google.  com
```

sep=" " ifadesi görünmezdir, yani aslında o arka planda çalışır ve default olarak tırnak içindeki ifadelerin arasında boşluk bırakır. Ancak tırnak içindeki ifadelerin arasına boşluk değil de başka bir karakter koymak istersek o zaman iş değişir. O halde yapmamız gereken sep parametresinin

içine istediğimiz karakteri koymaktır. O halde sep parametresine (yani sep= " " deki çift tırnak arasına) hiçbir şey yazmaz isek tırnak içindeki ifadelerin arasında hiç boşluk oluşmaz. O halde doğru kodumuzu yazalım.

print("www.", "google.", "com", sep="") **#Bir boşluk nelere kadir**
Çıktı: www.google.com

end parametresi: print() komutu içerisinde kullanılır. Yazdırılmak istenen ifadelerin sonuna hangi karakterin geleceğini belirler. Varsayılan olarak "\n" parametresi ile birlikte gelir. Yani yazılan ifade bitince bir alt satıra geçer.

Ör:
print(1)
print(2)
print(3)

Çıktısı:
1
2
3

#Yukarıda herhangi bir end parametresi göremiyoruz. Ancak Python yukarıdaki kodları aslında şu şekilde algılar:

print(1, end="\n")
print(2, end="\n")
print(3, end="\n")

#şimdi end parametresinin değerlerini değiştirelim. Her bir kod satırının sonuna tire (-) işareti koyalım.

print(1, end="-")
print(2, end="-")
print(3, end="-")

Çıktı: 1- 2- 3-

#Böylece yazdırılan ifadeler alt alta değil de yan yana gelecek şekilde ve aralarında tire(-) işareti olarak yazdırıldılar.