

Örnek-1:

```
1 ad=str(input("Adınız: "))
2 cins=str(input("Cinsiyetiniz (E/K): "))
3
4 if cins.upper()=="E":
5     print("Merhaba ", ad, " Bey")
6 else:
7     print("Merhaba ", ad, " Hanım")
8
```

- 1 ve 2. satırda kullanıcıdan string(karakter) tipinde veri alınıyor. Bu veriler 'ad' ve 'cins' değişkenlerine atanıyor.
- 4. satırda cins değişkeninin tuttuğu verinin "E"ye eşit olup olmadığı kontrol ediliyor. Büyük/küçük harf duyarlılığını ortadan kaldırmak için upper() metodu kullanılarak cins değişkeninin tuttuğu verinin büyük hali kontrole sokuluyor.
- 4. satırdaki şart sağlanırsa 5 numaralı satır, sağlanmazsa 7 numaralı satır çalışır.
- 6. satırdaki 'else:' terimi 4.satırdaki şartın sağlanmaması durumunda yapılacakları belirler.

Örnek-2:

```
1 ort=float(input("Puan Ortalaması:"))
2 if (ort>=85): print("Notunuz 5")
3 elif (ort>=70): print("Notunuz 4")
4 elif (ort>=60): print("Notunuz 3")
5 elif (ort>=50): print("Notunuz 2")
6 else: print("Notunuz 1")
7
```

- 1.satırda kullanıcıdan float(reel sayı) tipinde veri alınıyor ve 'ort' değişkenine atanıyor.
- 2.satırda ort değerinin 85'e eşit ya da 85'ten büyük olup olmadığı kontrol ediliyor ve şart sağlanırsa yapılacaklar belirtiliyor. (Ekran "Notunuz 5" yaz)
- 3-4-5.satırların her birinde kendisinden önceki satırdaki şartın sağlanmaması durumunda 'elif' terimiyle yeni kontrol gerçekleştiriliyor.
- 6. satırda, önceki kontrollerden hiçbiri sağlanmadığında yapılacaklar 'else:' ifadesi ile belirtiliyor.

Örnek-3:

```
1 for i in range(7):
2     print("Bugün haftanın", i+1, ".günü")
3
```

- Otomatik arttırımlı döngü, i değişkeninin değerini 0'dan 7'ye kadar (7 dahil değil) arttıracaktır.
- Döngü değişkeni olan i'nin aldığı her yeni değer için döngünün girintisine yazılan satır(lar) bir kez işletecektir.
- i'nin alacağı ilk değer 0 olduğu için gün sayısını belirtirken i+1 ifadesi kullanılmıştır.

Örnek-4:

```
1 for hafta in range(1,53):
2     for gun in range(1,8):
3         print("Günaydın, bugün", hafta, ".haftanın", gun, ".günü")
4 print("Bugün yılın son günü. MUTLU YILLAR...")
```

- Bu örnekte iç içe döngü kullanımı görülüyor. İçteki döngü, dıştaki döngünün her yeni tekrarı için tam tur çalışacaktır. Yani 'hafta' değişkeni 1 değerindeyken 'gun' değişkeni sırasıyla 1,2,3,4,5,6,7 değerlerini alacak ve her yeni değeri için 3.satırdaki kod çalışacaktır.
- Bu halde, 3.satırdaki kod 52x7=364 kez çalışacaktır.
- İç içe döngü yapısı tüm değerleri için çalıştıktan sonra döngüler durur ve 4.satırdaki kod bir kez çalışır.
- Böylece ekranda toplam 365 satır çıkış görünecektir.

Örnek-5:

```
1 top=0
2 for gun in range(30):
3     print(gun+1, ".günün hasılatını giriniz:")
4     gunluk=int(input(""))
5     top=top+gunluk
6     print("Bugüne kadar hasılat:", top, "TL")
7 print("Aylık Hasılatınız:", top, "TL")
```

- Bir değişkenin değeri kullanıcıdan alınmıyor ya da bir işlem sonucu ortaya çıkmıyorsa, bu değişkeni kullanabilmek için değişkene bir başlangıç değeri atanmalıdır. 1. satırda top değişkenine bu yüzden "0" değeri atanıyor. Bu atama yapılmıyorsa, 5. satırda "toplama işleminde "top" değişkeninin bir mevcut değeri olmadığı için değer hatası oluşacaktı.
- 4.satırda kullanıcıdan tamsayı değer alınıyor. Bu satır for döngüsü içinde olduğundan işlem 30 defa tekrarlanarak ayın her gününe ait hasılat ayrı ayrı elde ediliyor.
- 5. satırda ilk değeri "0" olan top değişkenine "gunluk" değişkenindeki değer ekleniyor. Bu satır da döngü içinde olduğundan 30 defa tekrar edilecek ve aylık hasılat "top" değişkeni içinde toplanacaktır. (Biriktirme)
- Elde edilen aylık hasılat değeri 7.satırda döngüden çıkıldıktan sonra ekrana yazdırılıyor.

Örnek-6:

```
1 say=0
2 while (say<100):
3     say=say+1
4     print(say)
```

- while, şarta bağlı çalışan döngüdür. Şart gerçekleştiği sürece yani "true" değeri ürettiği sürece döngü içindeki satırlar tekrar ederek işletilir.
- Bu örnekte "say" değişkeninin değeri 100'den küçük olduğu sürece döngü içindeki satırlar tekrar edecek, böylece ekrana 1'den 100'e kadar sayılar yazılacaktır.

- 3.satırda, başlangıç değeri "0" olan say değişkeninin değeri her tekrarda 1 artacak ve hemen ardından bu değer ekrana yazılacaktır.
- say değişkeninin değeri 100 olduğunda, artık 2. satırdaki şart sağlanmayacağından döngü devam etmez ve program akışı döngü dışında devam eder. Bu örnekte döngü dışında başka kod olmadığından program 1'den 100'e kadar sayıları yazar ve durur.

Örnek-7:

```
1 say=0
2 while True:
3     say=say+1
4     print(say)
5     if (say>=100):
6         break
```

- while döngüsünün çalışması için şartın sağlanması yani "true" değer üretmesi gerektiğini biliyoruz. Bundan dolayı bu örnekte 2. satırda while terimi yanına şart değil, doğrudan "true" yazılmış ve döngünün bu kısımda kontrol gerçekleştirilmeden tekrar etmesi sağlanmıştır.

- Bu örnek, 6. örnekle aynı işi yapmaktadır.
- Döngü satırında şart belirtilmediğinden döngü sürekli tekrar edecektir. Ancak her döngünün mutlaka beklenen durumda sonlanması gerekmektedir. Burada döngünün sonlanması için gerekli şart 5.satırda veriliyor.
- 5.satırdaki şart gerçekleştiğinde 6.satırdaki break komutu çalışacak ve döngü sonlandırılacaktır.

Örnek-8:

```
1 say=50
2 while (say>0):
3     bil=int(input("Satılan bilet sayısı: "))
4     say=say-bil
5     print("Kalan koltuk sayısı:", say)
6 print("Koltuk kalmadı lütfen bilet satmayınız.")
```

- say değişkeninin değeri "0"dan büyük olduğu sürece döngü çalışacak ve her turda 4.satırdaki eksiltme işlemi gerçekleşecektir. say değeri "0" olduğunda döngü sonlanacak ve 6.satırdaki metin yazılacaktır.

Örnek-9:

```
1 def say(x):
2     if x%2==0: print(x,"bir çift sayıdır.")
3     else: print(x,"bir tek sayıdır.")
4 def fonk(ad):
5     print("Merhaba "+ad)
6 print("Hoşgeldiniz")
7 isim=input("Adınız:")
8 fonk(isim)
9 sayi=int(input("Bir sayı giriniz:"))
10 say(sayi)
```

- 1. ve 4. satırlarda say ve fonk adlı iki fonksiyon tanımlanıyor. say fonksiyonunun çalıştıracağı kodlar 2 ve 3.satırlarda; fonk isimli fonksiyonun çalıştıracağı kodlar ise 5 satırda yer alıyor.
- Fonksiyonlar tanımlandıkları satırda değil, çağırılıkları satırda çalışırlar. Bu örnekte de fonk fonksiyonu 8. satırda, say fonksiyonu ise 10. satırda çağırılıyor.
- Fonksiyonlar çağırıldıklarında hangi değerle çalışacakları parantez içinde bildirilir. Örneğin, 10. satırda say fonksiyonuna "sayi" isimli değer gönderiliyor. "say(sayi)". Buradaki "sayi" değişkenin değeri, 1.satırda "x" olarak tanımlanan say fonksiyonunun parametresinin yerine geçerek 2 ve 3.satırlardaki işlemlere girecek.
- 2.satırdaki "%" karakteri bir işlem operatörüdür. MOD alma olarak bilinen, bölme işleminden kalanı bulan işlemdir. Burada $x\%2==0$ ifadesi ile x yerine gönderilen değerın 2'ye bölümünden kalanın olup olmadığı kontrol ediliyor.

Örnek-10:

```
1 def orta(a,b,c,d):
2     ort=(a+b+c+d)/4
3     print("Ders Ortalaması:",ort)
4 y1=int(input("1. yazılı notunuz:"))
5 y2=int(input("2. yazılı notunuz:"))
6 p1=int(input("1. performans notunuz:"))
7 p2=int(input("2. performans notunuz:"))
8 orta(y1,y2,p1,p2)
```

gönderilmiş.(y1,y2,p1,p2). Bu değerler sırasıyla (a,b,c,d) parametrelerinin yerine yerleşerek fonksiyonda hesaplama gerçekleştirilecektir.

Örnek-11:

```
1 def orta(a,b,c,d):
2     ort=(a+b+c+d)/4
3     return(ort)
4 y1=int(input("1. yazılı notunuz:"))
5 y2=int(input("2. yazılı notunuz:"))
6 p1=int(input("1. performans notunuz:"))
7 p2=int(input("2. performans notunuz:"))
8 x=orta(y1,y2,p1,p2)
9 son=x*x
10 print("Ortalama değerinin karesi:",son)
```

- Programlama etkinliklerinde fonksiyonların ortaya çıkardığı sonuçlar her zaman kullanıcıyla paylaşılmayabilir. Programcı fonksiyonun ortaya çıkardığı bir değeri başka bir işlemde kullanmak isteyebilir. Bu durumda değer ekrana yazdırılmaz fakat fonksiyonun çağırıldığı yere gönderilir. Değeri fonksiyonun çağırıldığı satıra göndermek için "return()" kullanılır.

- Bu örnekte programcı, kullanıcıdan aldığı değerlerin aritmetik ortalamasını orta fonksiyonuna hesaplatıyor fakat bu değeri kullanıcıya bildirmiyor. orta() fonksiyonu 8.satırda çağırıldığı için 3.satırdaki return(ort) ifadesi ortalama işleminin sonucunu 8.satırdaki yerine gönderiyor. Aynı satırda bu değer x değişkenine atanıyor ve 9. satırda başka bir işlemde girdi olarak kullanılıyor.

Uygulama sınavına hazırlanırken bu belgenin yanı sıra, daha önce aynı sayfada paylaşılan [ders notlarını](#) da mutlaka gözden geçiriniz.