

BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ PYTHON PROGRAMLAMA ÇALIŞMA NOTLARI

Tanımlanan değişkenlerde işlem önceliğini gözetme. Aritmetiksel, mantıksal ve atama operatörleri işlem önceliği.

Aritmetiksel operatörler, sayılar üzerinde matematiksel işlemler yapmak için kullanılır.

Operatör	Açıklama	Örnek	Sonuç
+	Toplama	3 + 2	5
-	Çıkarma	5 - 2	3
*	Çarpma	4 * 3	12
/	Bölme	10 / 2	5.0
%	Mod (kalan bulma)	10 % 3	1
**	Üs alma	2 ** 3	8
//	Tamsayı bölmesi (floordiv)	10 // 3	3

Mantıksal operatörler, iki veya daha fazla koşulu karşılaştırmak veya bir koşulun doğruluğunu kontrol etmek için kullanılır.

Genellikle koşul ifadelerinde ve karşılaştırmalarda kullanılır.

Operatör	Açıklama	Örnek	Sonuç
and	Tüm koşullar doğruysa True	True and False	False
or	Koşullardan biri doğruysa True	True or False	True
not	Doğruyu yanlış, yanlış doğruya çevirir	not True	False

Atama Operatörleri, bir değeri bir değişkene atamak veya mevcut bir değeri güncellemek için kullanılır.

Operatör	Açıklama	Örnek	Sonuç
=	Değişkene değer atar	x = 5	x = 5
+=	Değeri toplar ve sonucu atar	x += 3 (x = x + 3)	x = 8
-=	Değeri çıkarır ve sonucu atar	x -= 2 (x = x - 2)	x = 6
*=	Değeri çarpar ve sonucu atar	x *= 4 (x = x * 4)	x = 24
/=	Değeri böler ve sonucu atar	x /= 2 (x = x / 2)	x = 12.0
%=	Mod alır ve sonucu atar	x %= 5 (x = x % 5)	x = 2.0
**=	Üssünü alır ve sonucu atar	x **= 3 (x = x ** 3)	x = 8
//=	Tamsayı bölümünü alır ve sonucu atar	x //= 3 (x = x // 3)	x = 2

Python programlama dili karar yapılarını içeren programlar geliştirir.

KOŞULLU DURUMLAR: if, else, elif

if: "eğer" anlamı taşır. Bir koşulun sağlanması durumunda kullanılır.

else: "değilse" anlamı taşır. Bir koşulun sağlanmaması durumunda kullanılır.

#Klavyeden girilen Yaşa göre ehliyet alabilme sorgusu yapan program:

```
yas = int(input("Yaşınızı giriniz: ")) # sayısal bir veri kullandığımızdan dolayı int kullanmalıyız.
if yas < 18:
    print("Ehliyet alamazsınız!")
else:
    print("Ehliyet alabilirsiniz")
```

#klavyeden girilen Bir sayının tek mi çift mi olduğunu tespit eden program:

```
sayi=int(input("Bir sayı girin: "))
if sayi%2== 0:
    print("Girdiğiniz sayı çift sayıdır.")
else:
    print("Girdiğiniz sayı tek sayıdır.")
```

#Parola girilip girilmediğini denetleyen kodlar.

```
parola = input("Parolanızı oluřturunuz: ")
if not parola: #kullanıcı herhangi bir veri girmemiřse yani deęer false ise
print("Parola alanı boř bırakılamaz!")
else:
print("Parolanızı bařarıyla girdiniz! Parolanız: " , parola)
```

#Kullanıcı adı yazılmamıřsa uyarı veren program:

```
kullanici = input("Kullanıcı adınızı oluřturun: ")
if kullanıcı: #kullanıcı herhangi bir veri girmiřse yani deęer true ise
print("Kullanıcı adınız bařarıyla oluřturuldu! Kullanıcı adınız: ", kullanıcı)
else:
print("Kullanıcıdı boř bırakılamaz!")
```

#Parola uzunluęuna göre iřlem yapan program:parolanız 6-8 karakterden oluřmalıdır.

Parola=input()

Parolauzunluk=len(parola)

```
parola=input("Parolanız: ")
toplam_uzunluk=len(parola)
if toplam_uzunluk<6:
print("Parolanızın en az altı karakterden oluřmalı!")
else:
print("Hořgeldiniz!")
```

elif: "eęer deęilse" anlamı tařır. Eęer kodlarda sürekli if kullanırsak program tüm kořulları gözden geęirip ona göre karar verir. Ancak elif sadece bir önceki if bloęuna göre hareket eder.

#Boya göre yorum yazan program: boyunuz 160 cm den kısaysa minyon 180 cm den kısaysa normal 200 cm den kısaysa uzun 200 cm üstünde ise çok uzun uyarısı veren program

```
boy = int(input("boyunuz kaç cm?"))
if boy < 160:
    print("boyunuz minyon")
elif boy < 180:
    print("boyunuz normal")
elif boy < 200:
    print("boyunuz uzun")
else:
    print("boyunuz çok uzun")
```

Not: "if" ile "elif" arasındaki farkı daha iyi anlamak için yukarıdaki programda elif yazan yerleri silip yerine if yazın. Bu durumda boyunuz kaç cm sorusuna 155cm gibi bir cevap verseydik çıktı aşağıdaki gibi olacaktı. Çünkü "if" tüm koşulları değerlendirmiş olacaktı.

Çıktı:

boyunuz minyon

boyunuz normal

boyunuz uzun

Bir problemi çözmek için kullanacağı döngü yapısına karar verir.

Döngüler

İfade ve Kavram	Açıklama	Örnek
for döngüsü	for döngüleri belirli sayıda işlemlerin tekrarlanması için kullanılan döngülerdir. for döngüleri başlangıç ve bitiş değerleri arasında artım miktarına göre istenilen sayıda tekrar yapar.	for n in range(1,6): print(n)
range (başlangıç değeri, son değer, arttırma/azaltma değeri) :	Başlangıç değeri: Döngü değişkeninin alacağı ilk değerdir. Eğer boş bırakılırsa 0 olarak belirlenir. Son değer: Döngü değişkeninin bitiş değeridir. Boş bırakılmamalıdır. Arttırma/azaltma değeri: Döngü değişkeninin arttırma veya azaltma miktarını belirler. Eğer boş bırakılırsa, 1 olarak belirlenir.	range(10) → 0,1,2,3,4,5,6,7,8,9 range(1, 10) → 1,2,3,4,5,6,7,8,9 range(1, 10, 2) → 1,3,5,7,9 range(10, 0, -1) → 10,9,8,7,6,5,4,3,2,1 range(10, 0, -2) → 10,8,6,4,2 range(2, 11, 2) → 2,4,6,8,10 range(-5, 5) → -5,-4,-3,-2,-1,0,1,2,3,4 range(1, 2) → 1

ÖRNEK 1.

```
for n in range(1,11):  
    print(n)
```

ÖRNEK 2.

```
for n in range(1,11,2):  
    print(n)
```

ÖRNEK 3.

```
for n in range(20,0,-2):  
    print(n)
```

ÖRNEK 4.

```
for n in range(30):  
    print(n)
```

ÖRNEK 5.

```
for a in range(5):  
    print("Merhaba Dünya")
```

ÖRNEK 6.

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

ÖRNEK 7.

1 ile 100 arasındaki sayıları toplayan programı yazınız.

```
toplam= 0  
for i in range(1, 100):  
    toplam+= i  
print(toplam)
```

ÖRNEK 8. (10-c) (10-A)

1 ile 10 arasındaki sayıların karesini alan programı yazınız.

```
for sayi in range(1,11):  
    kare=sayi**2  
    print("Sayı : ",sayi,"Karesi : ",kare)
```

İfade ve Kavram	Açıklama	Örnek
while döngüsü	While ifadesi, ilgili kod satırlarının çalıştırılıp çalıştırılmayacağını belirler. Şart doğru olduğu sürece kod bloğunu tekrar tekrar çalıştırır. Şart yanlış olduğunda ise döngü sonlanır. Türkçede '... iken, ... olduğu sürece' gibi anlamlarına gelir.	n = 1 while n <= 10: print(n) n += 1
Sonsuz döngü	Sonsuz döngü çeşitli sebeplerle sonsuza kadar döngüye giren ve sonlanamayan bilgisayar programı komutu parçalarına verilen addır. Bu sebepler döngünün; bir sonlandırıcı koşulun bulunmaması, bulunsa da hiçbir zaman sağlanamayacak olması ya da bu koşulun döngünün her seferinde yeniden başlamasına neden olması olabilir.	a = 1 while a < 10: print("Ali")
while True:	Aksi belirtilmediği sürece programı çalıştırır. Bu haliyle programımızı çalıştırsak sonsuz döngüye girmiş oluruz. Ancak ; break komutunu ilave edersek programa dur komutu vermiş oluruz.	while True: print("Bilgisayar çıldırdı!") break
Belirli Döngü	Döngünün kaç defa döneceği belirlidir.	n = 1 while n <= 10: print(n) n += 1
Belirsiz Döngü:	Döngünün kaç defa döneceği belirsizdir. Döngünün kaç defa döneceğine şartlar ve kullanıcı karar verir.	n = 1 karar= int(input("sayılar kaçça kadar sıralansın?")) while n <= karar: print(n) n += 1

Örnek 1:

Ekrana 10 kez ali yazdıran programı while döngüsü kullanarak yazınız.

```
a=0
```

```
while a<10:
```

```
  print("ali", end=" ")
```

```
  a+=1
```

Örnek 2: Sonsuz döngü örneği (Ctrl+c ile sonsuz döngüden çıkılır)

```
a=1
```

```
while a<2:
```

```
  print("merhaba dünya")
```

```
  break :                           #sonsuz döngü kırılır.
```

Örnek 3: 1 ile 50 arasında üçe bölünebilen sayıları ekrana while döngüsü ile yazdırınız

```
a=1
while a<50:
    a+=1
    if a%3==0:
        print(a, end=" ")
```

Örnek 4: Klavyeden girilen iki sayı arasındaki sayıları toplayan programı yazınız?

```
top=0
ilk=int(input("başlangıç değerini giriniz ="))
son=int(input("son değerini giriniz ="))
for i in range(ilk,son+1):
    top+=i
print(top,sep="
```

.....

```
top=0
while True:
    ilk=int(input("başlangıç değerini giriniz ="))
    son=int(input("son değerini giriniz ="))
    if son<ilk:
        print("ilk değeri küçük gimelisiniz")
        continue
    while ilk<son:
        for i in range(ilk,son+1):
            top+=i
        print(top,sep=" ")
        break
    break
```

Örnek 5: 0 dan başlayarak kullanıcının girdiği sayıya kadar olan sayıları 1 artırarak ekranda gösteren programı while döngüsünü kullanarak yazınız. kullanıcı sayıyı negatif girerse uyarı versin.

```
ilk=0
sayı=int(input("sayı gir="))
while sayı<0:
    sayı=int(input("sayı gir="))
    if sayı<0:
        print("lütfen sıfırdan büyük sayı giriniz")
    continue
while ilk<sayı:
    print(ilk,end=" ")
    ilk+=1
break
```

İfade ve Kavram	Açıklama	Örnek
İç içe döngüler	Bir döngü yapısının içine başka bir döngü yapısının yerleştirilmesi ile elde edilen yapıdır. İç içe döngülerde önce içteki çevrimler tamamlanması gerektiğidir. İç içe döngülerde en içteki çevrim en önce tamamlanır.	for i in range(1,4): print(i) for j in range(1,4): print(j)
break ifadesi	Döngüyü sonlandıran bir komuttur.	
continue ifadesi	continue komutunun görevi kendisinden sonra gelen her şeyin es geçilip döngünün başa dönmesini sağlamaktır.	
pass ifadesi	Herhangi bir işlem yapmadan geçeceğimiz durumlarda kullanılır. Kısaca "Hiçbir şey yapmadan yola devam et!" anlamı katar.	

Örnek 1: (bir a yazıp 3 b yazar dıştaki döngü kadar tekrar eder.)

```
for i in range(3):
    print("a")
for j in range(3):
    print("b")
```


Örnek2: continue kendinden sonra gelen komutları es geçip döngüyü başa döndürür.

```
for i in range(2):  
    print("a")  
    continue  
    for j in range(3):  
        print("b")
```

Örnek 3: break ifadesi görüldüğünde döngü sonlanır.

```
for i in range(2):  
    print("a")  
    break  
    for j in range(3):  
        print("b")
```

Örnek 4: Dıştan gelen sayı içteki döngünün tamamıyla çarpılır.

```
for satir in range(1,5):  
    for sütun in range(1,5):  
        print(satir*sütun,end=" ")
```