

PYTHONDA KULLANILAN TEMEL İFADELER VE DÖNGÜLER

```
In [2]: #if - elif - else
renk= input ('Şu renklerden birisini girin- kırmızı,turuncu veya mavi')
if renk== 'kırmızı':
    print('Favori renginiz kırmızı')
elif renk == 'turuncu':
    print('Favori renginiz turuncu')
elif renk == 'mavi' :
    print ('En sevdiğiniz renk mavi')
else: print ("Yanlış rengi girdiniz")
```

En sevdiğiniz renk mavi

```
In [3]: x=20
if x<10:
    if x<5:
        print("5'ten küçük sayı")
    else:
        print("5'ten büyük sayı")
else:
    print("10'dan büyük sayı")
```

10'dan büyük sayı

```
In [4]: a=200
b=300
c=500
if a<b and c>a:
    print("Her iki koşul da Doğru")
```

Her iki koşul da Doğru

```
In [5]: a=200
b=33
c=500
if a>b or a>c :
    print("Koşullardan en az biri Doğru")
```

Koşullardan en az biri Doğru

```
In [28]: a=200
b=33
if b>a:
    print("b, a'dan büyüktür")
else:
    print("b, a'dan büyük değildir")
```

b, a'dan büyük değildir

"FOR" DÖNGÜSÜ

Python'daki "for" ifadesi, herhangi bir dizinin öğeleri üzerinde sırayla yinelenerek tanımlanan işlemi uygular. Yani "for" ifadesi dizi üzerinde tanımlanan işlevi uygulayan bir döngü oluşturur. Bu dizi; liste, demet,sözlük veya küme olabilir. Örneğin; meyveler adında tanımlanan bir dizi için "print()"

işlevi her bir öge için sonuç verecektir.

```
In [31]: meyveler= ["elma", "muz", "kiraz"]
for x in meyveler:
    print(x)
```

elma
muz
kiraz

"For" döngüsünde "break" ifadesiyle, tüm öğeler arasında döngü oluşturmadan önce döngü durdurulabilir. Döngüde işlenmesi istenen işlevden sonra "break" tanımlanırsa dizi üzerinde belirtilen nesnede dahil edilerek döngü bitirilir.

```
In [33]: meyveler= ["elma", "muz", "kiraz"]
for x in meyveler:
    print(x)
    if x=="muz":
        break
```

elma
muz

Aşağıdaki örnekte x değeri "muz" olana kadar print(x) komutu çalışacaktır. Her bir satırın yazılmaya başladığı konuma dikkat ediniz.

```
In [39]: meyveler= ["elma", "muz", "kiraz"]
for x in meyveler:
    if x=="muz":
        break
    print(x)
```

elma

"continue" ifadesi dizi içerisinde belirtilen öge üzerinde önce kesinti oluşturur daha sonra döngünün devam etmesini sağlar.

```
In [43]: meyveler=["elma", "muz", "kiraz"]
for x in meyveler:
    if x=="muz":
        continue
    print(x)
```

elma
kiraz

"range()" DÖNGÜSÜ

Bir dizi kodda belirli sayıda döngü yapmak için "range()" işlevi kullanılabilir.

```
In [46]: for x in range(6):
          print(x)
```

```
0  
1  
2  
3  
4  
5
```

```
In [48]: for x in range (2,6):  
         print(x)
```

```
2  
3  
4  
5
```

```
In [50]: for x in range(2,30,3):  
         print(x)
```

```
2  
5  
8  
11  
14  
17  
20  
23  
26  
29
```

"for" Döngüsünde "else" İfadesi

Bir "for" döngüsündeki "else" ifadesi döngü bittiğinde yürütülecek işlevi belirtir.

0'dan 5'e kadar tüm sayıları ve döngü bittiğinde bir mesaj yazdıran uygulama aşağıdaki gibidir:

```
In [53]: for x in range(6):  
         print(x)  
         else:  
             print("Sonunda bitti!")
```

```
0  
1  
2  
3  
4  
5  
Sonunda bitti!
```

Yuvalanmış döngü, bir döngü içindeki belirtilen diğer bir döngüdür. "İç döngü", "dış döngünün" her yinelemesi için bir kez yürütülür.

Belirtilen diziler üzerinde kartezyen çarpım işlemi şeklinde sonuç verir.

```
In [58]: sıfat=["kırmızı", "büyük", "lezzetli"]
```

```
meyveler=["elma", "muz", "kiraz"]  
for x in sıfat:  
    for y in meyveler:  
        print(x,y)
```

```
kırmızı elma  
kırmızı muz  
kırmızı kiraz  
büyük elma  
büyük muz  
büyük kiraz  
lezzetli elma  
lezzetli muz  
lezzetli kiraz
```

"while" DÖNGÜSÜ

"while" döngüsü, bir koşul doğru olduğu sürece belirli komutları yürütmek için kullanılır ve koşul doğru olduğu sürece çalışmaya devam eder.

```
In [64]: i=1  
while i<6:  
    print(i)  
    i+=1
```

```
1  
2  
3  
4  
5
```

"continue" ifadesiyle mevcut yinelemeyi durdurabilir ve dizide yer alan bir sonraki öge ile uygulamaya devam edebiliriz.

```
In [5]: i=0  
while i<6:  
    i+=1  
    if i==3:  
        continue  
    print(i)
```

```
1  
2  
4  
5  
6
```

"else" ifadesiyle koşul artık doğru olmadığından tanımlanan işlev uygulanabilir. Koşul yanlış olduğunda tanımlanan mesajın yazdırılması aşağıdaki gibidir:

```
In [9]: i=1  
while i<6:  
    print(i)  
    i+=1  
else:
```

```
print("i artık 6'dan küçük değildir")
```

```
1
2
3
4
5
i artık 6'dan küçük değildir
```

"def" İFADESİ

"def" ifadesi kullanıcı tarafından tanımlanan fonksiyonlar olarak tanımlanabilir.

"def" ifadesinde belirtilen fonksiyon yalnızca çağırıldığında çalışan bir kod bloğudur. Fonksiyon ilgili eylemi gerçekleştirmek için yeniden kullanılabilir bir yapıya sahiptir.

```
In [21]: def fonksiyonum(Adı):
        print(Adı+ " Çetin")
```

```
In [23]: fonksiyonum("Ahmet")
        fonksiyonum("Mehmet")
```

```
Ahmet Çetin
Mehmet Çetin
```

```
In [25]: def fonksiyonum2(yemek):
        for x in yemek:
            print(x)
```

```
In [27]: meyveler=["elma", "muz", "kiraz"]
```

```
In [29]: fonksiyonum2(meyveler)
```

```
elma
muz
kiraz
```