

İç İçe Karşılaştırma İşlemleri

Tek bir **if deyimi**, programımızın iki seçenektan birini seçmesine olanak sağlar. Ama pratikte daha fazla seçeneğe ihtiyaç duyan problemlerle karşılaşabiliriz.

Birinci karardan sonra ikinci, ikinciden sonra üçüncü kararın alınması gerekebilir. Bu şekilde giden bir program akışı için **İÇ İÇE if deyimleri** kullanmamız gerekiyor.

Aşağıdaki örnekte suyun sıcaklık değerine göre karar verilmesi gereken üç durum var ve **düz mantıkla** yani her bir olasılık için birer tane if deyimi kullanılmış.

Kaynak Kod 1	
1	derece = int(input("Suyun sıcaklığını giriniz :"))
2	if derece < 0 :
3	print ("Katı halde ")
4	if derece >=0 and derece<100:
5	print ("Sıvı halde ")
6	
7	if derece >=100:
8	print ("Gaz halde ")

Bu şekilde düz mantık kullanmak yerine aşağıda olduğu gibi ilk önce if deyimi ile derecenin 0'dan küçük olup olmadığına, küçük değilse 100'den küçük olup olmadığına bakılabilir.

Kaynak Kod 2	
1	derece = int(input("Sıcaklığını giriniz :"))
2	if derece < 0 :
3	print ("Katı halde ")
4	else:
5	if derece <100:
6	print ("Sıvı halde ")
7	else:
8	print ("Gaz halde ")

Aşağıdaki örnek kodları yazıp programı çalıştırdığımızda "elma" ve "karpuz" haricinde girilen bilgiler için gayet mantıklı çalışıyormuş gibi gelebilir. Ama "elma" girersek hem elmanın bir meyve olduğunu yazacak hem de "elma gerçekten bir meyve midir?" diye soracaktır. **Bunun nedeni örneğimizdeki üç if deyiminin birbirinden bağımsız olarak karşılaştırma yapmasıdır.**

Kaynak Kod 3	
1	cevap = input("Bir meyve adı söyleyin bana: ")
2	if cevap == "elma" :
3	print("evet, elma bir meyvedir...")
4	if cevap == "karpuz" :
5	print("evet, karpuz bir meyvedir...")
6	if cevap == "armut" :
7	print("evet, armut bir meyvedir...")
8	else :
9	print(cevap, "gerçekten bir meyve midir?")

3. Örneği aşağıdaki gibi iç içe if deyimi kullanarak yazarsak programımızdaki mantık hatasını gidermiş oluruz. Her yazılan if deyimi öncekinin içersinde yer alıyor. İlk karşılaştırma doğru olduğu zaman "evet, elma bir meyvedir..." yazıp program bitiyor. Yanlış ise bu sefer else bloğundaki if deyiminin karşılaştırması yapılıyor. Doğru ise "evet, karpuz bir meyvedir..." yazıp program bitiyor. Bu şekilde en son if deyimindeki karşılaştırma sonucuda yanlış çıkarsa kullanıcının girdiği cevabın gerçekten bir meyve olup olmadığı soruluyor.

Kaynak Kod 4	
1	cevap = input("Bir meyve adı söyleyin bana:")
2	if cevap == "elma":
3	print("evet, elma bir meyvedir...")
4	else:
5	if cevap == "karpuz":
6	print("evet, karpuz bir meyvedir...")
7	else:
8	if cevap == "armut":
9	print("evet, armut bir meyvedir...")
10	else:
11	print(cevap, "gerçekten bir meyve midir?")

elif deyimi – Zincirleme Karşılaştırma

İç içe karşılaştırmalarda her seferinde else bloğu ve onun içerisinde if bloğu şeklinde yazmak yerine Python diline özgü else – if'in kısaltması olan **elif deyimi** kullanılır. Bu deyimi istediğimiz sayıda kullanabiliriz. Karşılaştırma işlemi if deyiminden başlayarak sonucu doğru çıkan ilk koşula ait blok işletilir. Hiçbiri doğru çıkmaz ise en sondaki else bloğu işletilir.

Kaynak Kod 5	
1	cevap = input("Bir meyve adı söyleyin bana:")
2	if cevap == "elma":
3	print("evet, elma bir meyvedir...")
4	elif cevap == "karpuz":
5	print("evet, karpuz bir meyvedir...")
6	elif cevap == "armut":
7	print("evet, armut bir meyvedir...")
8	else:
9	print(cevap, "gerçekten bir meyve midir?")

Aşağıdaki örnek; kullanıcının girdiği üç kenar uzunluğuna göre üçgenin türünü belirliyor.

Kaynak Kod 6	
1	a = int(input("1. Kenar uzunluğu: "))
2	b = int(input("2. Kenar uzunluğu: "))
3	c = int(input("3. Kenar uzunluğu: "))
4	if a==b and b==c:
5	print("Eş kenar üçgen")
6	elif a==b or b==c:
7	print("İkiz kenar üçgen")
8	else:
9	print("Çeşit kenar üçgen")