

Döngülerde kullanılan ifadeler : break ve continue

Bu videoda döngülerde kullanılabilen *break* ve *continue* ifadelerini öğrenmeye çalışacağız. Bu ifadeleri kullanarak döngü yapılarını daha efektif kullanabiliriz.

break ifadesi

break ifadesi döngülerde programcılar tarafından en çok kullanılan ifadedir. Anlamı şu şekildedir;

Döngü herhangi bir yerde ve herhangi bir zamanda break ifadesiyle karşılaştığı zaman çalışmasını bir anda durdurur. Böylelikle döngü hiçbir koşula bağlı kalmadan sonlanmış olur.

break ifadesi **sadece ve sadece** içindeki bulunduğu döngüyü sonlandırır. Eğer iç içe döngüler bulunuyorsa ve en içteki döngüde break kullanılmışsa sadece içteki döngü sona erer. Örneklerle *break* ifadesini anlamaya çalışalım.

```
In [2]: i = 0 # Bu döngüyü biliyoruz.
```

```
while (i < 20):  
    print(i)  
    i +=1
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19
```

In [5]: `i = 0 # break kullanmaya çalışalım.`

```
while (i < 20):
    print(i)
    if (i == 10):
        break # i'nin değeri 10 olunca bu koşul sağlanıyor ve break ifadesiyle karşı
        karşılaştığı için döngü anında sona eriyor.
    i +=1
```

0
1
2
3
4
5
6
7
8
9
10

In [3]: `# for döngüsüyle break kullanalım.`

```
liste = [1,2,3,4,5,6,7,8,9]
for i in liste:
    if (i == 5):

        break
    print(i)
```

1
2
3
4

In [2]: `while True: # Sonsuz döngü. Nasıl sonlandırabiliriz ?`

```
    isim = input("İsminiz(Çıkmak için q tuşuna basın.):")
    if (isim == "q"): # break ile tabii ki.
        print("Çıkış yapılıyor...")
        break
    print(isim)
```

İsminiz(Çıkmak için q tuşuna basın.):q
Çıkış yapılıyor...

İşte `break` ifadesi bu kadar ! Şimdi de `continue` ifadesine bakalım.

continue ifadesi

`continue` ifadesi `break`'e göre biraz daha az kullanılan bir ifadedir. Anlamı şu şekildedir;

Döngü herhangi bir yerde ve herhangi bir zamanda `continue` ifadesiyle karşılaştığı zaman geri kalan işlemlerini yapmadan direkt bloğunun başına döner.

`continue` ifadesini anlamak için örneklerimize bakalım.

```
In [7]: liste = [1,2,3,4,5,6,7,8,9,10]

for i in liste: # Bunu biliyoruz.
    print("i:",i)
```

```
i: 1
i: 2
i: 3
i: 4
i: 5
i: 6
i: 7
i: 8
i: 9
i: 10
```

```
In [6]: liste = [1,2,3,4,5,6,7,8,9] # continue kullanalım.

for i in liste:
    if (i == 3 or i == 5):
        continue
    print("i:",i)
```

```
i: 1
i: 2
i: 4
i: 6
i: 7
i: 8
i: 9
```

continue ifadesi while döngüleriyle kullanıldıkları zaman **sonsuz döngü** olayına yol açabilmektedir. Anlamak için örneğimize bakalım.

Jupyterde sonsuz döngüyü çalıştırdığımızda , programı durdurmak için ilk olarak yukarıdaki "Kernel" e gidelim ve Interrupta tıklayalım. Bir kaç saniye bekledikten sonra "Restart" a tıklayalım. Böyle bir durumda Jupyter tekrar kendine gelecektir.

```
In [ ]: i = 0 # Bu kodda Sonsuz döngü olayı neden oluşur ? Bu kodu çalıştırmayalım.

# Eğer çalıştırsak sonsuz döngüyü "Kernel" sekmesinde

while (i < 10):

    if (i == 2):
        continue

    print(i)
    i += 1
```

```
In [9]: i = 0 # Kodun sorunsuz hali

while (i < 10):

    if (i == 2):
        i += 1 # Artırma işlemi
        continue

    print("i:",i)
    i += 1
```

```
i: 0
i: 1
i: 3
i: 4
i: 5
i: 6
i: 7
i: 8
i: 9
```

İşte *break* ve *continue* ifadeleri bu şekillerde kullanılmaktadır. Kısım sonundaki alıştırmalarla bu konuyu daha iyi kavrayacağız.

In []: