

Fonksiyonlarda Return

Bu konuda fonksiyonlardan değer döndürmemizi sağlayan **return** ifadesini görmeye çalışacağız. Önceki bölümde yazdığımız fonksiyonları hatırlayacak olursak, bu fonksiyonlar sadece ekrana **print** ile değer yazdırıyordu. Ancak bu fonksiyonlar yaptıkları işlemler bize herhangi bir değer vermiyorlardı. Ancak biz programlarımızda bir fonksiyon sonucunda elde edilen değerleri alıp programlarımızın bambaşka yerlerinde kullanmak isteyebiliriz. Bu derste bunu nasıl yapacağımızı öğrenmeye çalışacağız.

return ifadesi fonksiyonun işlemi bittikten sonra **çağrıldığı yere** değer döndürmesi anlamı taşır. Böylelikle, fonksiyonda aldığımız değeri bir değişkende depolayabilir ve değeri programın başka yerlerinde kullanabiliriz. Şimdi iki tane çok basit fonksiyon yazalım ve **return** neden gereklidir anlamaya çalışalım.

```
In [10]: def toplama(a,b,c): # Birinci fonksiyon
          print("Toplamları",a+b+c)

          def ikiyleçarp(a): # İkinci fonksiyon
              print("2 ile çarpılmış hali", a * 2)
```

```
In [11]: toplam = toplama(3,4,5)

         ikiyleçarp(toplam)
```

Toplamları 12

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-11-da4bfe0bd65f> in <module>()
      1 toplam = toplama(3,4,5)
      2
----> 3 ikiyleçarp(toplam)

<ipython-input-10-5128afde7d6b> in ikiyleçarp(a)
      3
      4 def ikiyleçarp(a): # İkinci fonksiyon
----> 5     print("2 ile çarpılmış hali", a * 2)
      6
```

TypeError: unsupported operand type(s) for *: 'NoneType' and 'int'

Burada hata almamızın sebebi fonksiyonları herhangi bir değer döndürmemesi yani **return** kullanmamasıdır. İsterseniz **toplam** değişkeninin tipine bakalım.

```
In [12]: type(toplam)
```

Out[12]: NoneType

Burada **toplama** fonksiyonundan herhangi bir değer döndürülmediği için **toplam** değişkenimiz hiçbir değere sahip olmadı ve tipi **NoneType**(atanmamış) olmuş oldu. İsterseniz burada fonksiyonları tekrardan tanımlayalım ve **return** mantığını anlamaya çalışalım.

```
In [13]: def toplama(a,b,c):
          return a+b+c # return'un kullanımı
          def ikiyle_çarp(a):
              return a*2
```

```
In [14]: toplam = toplama(3,4,5)
         print(ikiyle_çarp(toplam))
```

24

İşte **return** ifadesinin anlamı tam olarak budur. **return** yardımıyla fonksiyonlar değerleri çağrıldığı yere döndürebilir ve biz de bu değerleri istediğimiz yerde kullanabiliriz.

Örneğin , siz yemek yapacaksınız ve yemek için **soğanların kıyılması** gerekiyor. Bunun için **soğanları(değerleri)**,mutfak robotuna(fonksiyon) veriyorsunuz ve mutfak robotu kıyma işlemini gerçekleştiriyor. Ancak işlemlerini bitirdikten sonra mutfak robotu (ütopik bir örnek olsun) size soğanların kıyılmış halini vermiyor(**return** kullanmıyor). Sonuçta da siz bu kıyılmış soğanları alıp yemeğinizde kullanamıyorsunuz. Fonksiyonlarda da **returnun** anlamı budur. Şimdi bir tane daha örnek yapalım ve **returnu** iyice anlamaya çalışalım.

```
In [32]: def üçleçarp(a):
         print("1.fonksiyon çalıştı")
         return a*3
```

```
In [33]: def ikiyletopla(a):
         print("2.fonksiyon çalıştı")
         return a + 2
```

```
In [34]: def dördeböl(a):
         print("3.fonksiyon çalıştı")
         return a / 4
```

```
In [35]: # 3 ünü beraber kullanalım.

         print(dördeböl(ikiyletopla(üçleçarp(5))))

1.fonksiyon çalıştı
2.fonksiyon çalıştı
3.fonksiyon çalıştı
4.25
```

return ifadesinden sonra fonksiyonumuz tamamıyla sona erer. Yani, **return** ifadesinden sonra yapılan herhangi bir işlem çalıştırılmaz.

```
In [38]: def toplama(a,b,c):
         return a + b + c
         print("Toplama fonksiyonu") # Çalıştırılmadı.
```

```
In [39]: print(toplama(1,2,3))
```

6

```
In [40]: def toplama(a,b,c):
         print("Toplama fonksiyonu") # Çalıştırıldı.
         return a + b + c
```

In [41]: toplama(1,2,3)

Toplama fonksiyonu

Out[41]: 6

Fonksiyonlarda çağrıldığı yere herhangi bir değer döndürmeyen(**return kullanılmayan**) fonksiyonlara **void** fonksiyonlar denmektedir. Bunu da genel kültür olarak bilmekte fayda var.

Bir sonraki konuda **Fonksiyonlarda Parametre Türlerini** öğrenmeye çalışacağız.

In []: