

Decorator prensibi

@decorator def ...

Değişken olarak fonksiyon

```
In [1]: def deneme():  
        print("abc")
```

```
In [2]: deneme()
```

abc

```
In [3]: f=deneme
```

```
In [4]: f()
```

abc

```
In [5]: print(f)
```

<function deneme at 0x7fcd486ce050>

```
In [7]: print(f())
```

abc
None

Fonksiyon içinde fonksiyon

```
In [9]: def deneme():  
        print("Deneme fonksiyonu çalışıyor.")  
  
        def test(): #indentation gerekir ki altına bağlansın  
            return "Test fonksiyonu çalışıyor."  
  
        print(test())
```

```
In [10]: deneme()
```

Deneme fonksiyonu çalışıyor.
Test fonksiyonu çalışıyor.

```
In [24]: def deneme():  
        print("Deneme fonksiyonu çalışıyor.")  
  
        def test():  
            return "Test fonksiyonu çalışıyor."  
  
        return test() # döndürür bastırmaz
```

```
In [25]: deneme()
```

Deneme fonksiyonu çalışıyor.

Out [25]: 'Test fonksiyonu çalışıyor.'

```
In [26]: def deneme():  
        print("Deneme fonksiyonu çalışıyor.")  
  
        def test():  
            print("Test fonksiyonu çalışıyor.") #böyle basar  
  
        return test() #print ile böyle basar
```

```
In [27]: deneme()
```

Deneme fonksiyonu çalışıyor.
Test fonksiyonu çalışıyor.

```
In [28]: def deneme():  
        print("Deneme fonksiyonu çalışıyor.")  
  
        def test():  
            print("Test fonksiyonu çalışıyor.")  
  
        print(test()) #print içine print konmaz
```

In [29]: `deneme()` # none verir

Deneme fonksiyonu çalışıyor.
Test fonksiyonu çalışıyor.
None

In [35]:

```
def deneme():  
    print("Deneme fonksiyonu çalışıyor.")  
  
    def test():  
        return "Test fonksiyonu çalışıyor."  
  
    return test #böyle döndürmez bile ama...
```

In [36]: `deneme()`

Deneme fonksiyonu çalışıyor.

Out [36]: `<function __main__.deneme.<locals>.test()>`

In [37]:

```
f=deneme()  
f() #böyle döndürür
```

Deneme fonksiyonu çalışıyor.

Out [37]: `'Test fonksiyonu çalışıyor.'`

In [39]: `print(f())` # böyle sonuncuyu bastı

Test fonksiyonu çalışıyor.

In [42]: `print(f)` #böyle dönmeyenin sonuncusunu verdi

`<function deneme.<locals>.test at 0x7fcd58ddcaf0>`

Decorator

In [43]:

```
def deneme():  
    return "Deneme fonksiyonu çalışıyor."
```

In [48]:

```
def ikinci(f):  
    print("İkinci fonksiyon çalışıyor.")  
    print(f)
```

In [49]: `ikinci(deneme())`

İkinci fonksiyon çalışıyor.
Deneme fonksiyonu çalışıyor.

In [50]:

```
def ikinci(f):  
    print("İkinci fonksiyon çalışıyor.")  
    print(f()) # eğer içine halihazırda () koyarsan...
```

In [51]: `ikinci(deneme)` # burda koyman gerekmez çünkü zaten koydun

İkinci fonksiyon çalışıyor.
Deneme fonksiyonu çalışıyor.

In [56]:

```
def deco(f): # daha sonra tanımlanacak olan f e eşitlenecek fonksiyonu içine yazıyoruz  
  
    def wrapper():  
        print("Başlangıç.")  
  
        f() # burada f in nerede çalışacağını belirtiyoruz  
  
        print("Bitiş")  
  
    return wrapper
```

In [57]:

```
def yazdir():  
    print("Yazdir.")  
yazdir = deco(yazdir) # f yerine daha sonra tanımladığımız yazdir fonksiyonunu koyuyoruz ve artık wrappera tanımlanıyor
```

In [58]: `yazdir()` # böylece yazdir her çalıştığında deco içindeki wrapper ile çalışacak

Başlangıç.
Yazdir.
Bitiş

In [59]:

```
def yazdir2():  
    print("Yazdir2.")  
yazdir2 = deco(yazdir2)
```

```
In [60]: yazdir2()

Başlangıç.
Yazdır2.
Bitiş

In [66]: @deco # her seferinde decoyu baştan çağırmaya gerek yok bi kere yazdıktan sonra böyle çağrılabilir
def yazdir(): # indent yok
    print("yazdir")

In [67]: yazdir()

Başlangıç.
yazdir
Bitiş

In [68]: @deco
def toplama(a,b): # argüman alan fonksiyonlarda decorator çalışmaz
    print(a,b)

In [69]: toplama(4,5)

-----TypeError
-----> 1 toplama(4,5)
TypeError: deco.<locals>.wrapper() takes 0 positional arguments but 2 were given
Traceback (most recent call last)Cell In[69], 1
```

Argüman alan fonksiyonlarda decorator

```
In [70]: def deco(f):

    def wrapper(*args):
        print("Başlangıç.")

        f(*args)

        print("Bitiş.")

    return wrapper

In [71]: @deco
def toplama(a,b):
    print(a+b)

In [72]: toplama(4,5)

Başlangıç.
9
Bitiş.
```

Argüman alan decorator

```
In [82]: def deco(f, msg1, msg2):
           #kabul etmeyecek

    def wrapper(*args):
        print(msg1)

        f(*args)

        print(msg2)

    return wrapper

In [83]: @deco("Başlangıç", "Bitiş.")
def toplama(a,b):
    print(a+b)

-----TypeError
-----> 1 @deco("Başlangıç", "Bitiş.")
      2 def toplama(a,b):
      3     print(a+b)
TypeError: deco() missing 1 required positional argument: 'msg2'
Traceback (most recent call last)Cell In[83], 1

In [84]: def deco(msg1,msg2):
           #bir ara katman tanımladık böyle kabul edecektir

    def ara_katman(f):
        def wrapper(*args):
            print(msg1)

            f(*args)

            print(msg2)
```

```
return wrapper
return ara_katman #
```

```
In [85]: @deco("Başlangıç", "Bitiş.")
def toplama(a,b):
    print(a+b)
```

```
In [86]: toplama(4,5)
```

```
Başlangıç
9
Bitiş.
```

Örnek: Süre Ölçümlemesi

```
In [87]: import time
baslangic=time.time()
f()
bitis=time.time()
bitis-baslangic
```

```
Out [87]: 0.0001609325408935547
```

```
In [93]: def sure_olc(f):
def wrapper(*args):

    baslangic=time.time()
    print("Başlangıç zamanı:\t{}".format(baslangic))

    f(*args)

    bitis=time.time()

    print("Bitiş zamanı:\t{}".format(bitis))
    print("Geçen zamanı:\t{}".format(bitis-baslangic))
    return wrapper
```

```
In [94]: def faktoriyel(sayi):

    toplam=1

    while sayi > 1:
        toplam=toplam*sayi
        sayi-=1
    print(toplam)
```

```
In [95]: faktoriyel(5)
```

```
120
```

```
In [48]: @sure_olc
def faktoriyel(sayi):

    toplam=1

    while sayi > 1:
        toplam=toplam*sayi
        sayi-=1
    print(toplam)
```

```
In [49]: faktoriyel(6)
```

```
Başlangıç zamanı:      1708696694.299205
720
Bitiş zamanı:   1708696694.2992609
Geçen zamanı:   5.5789947509765625e-05
```

```
In [50]: faktoriyel(50)
```

```
Başlangıç zamanı:      1708696698.975334
40238726007709377354370243392300398571937486421071463254379991042993851239862902059204420848696940480047998861019719605863166687299480855890132382966994459099742450
Bitiş zamanı:   1708696698.9760032
Geçen zamanı:   0.0006692409515380859
```

```
In [52]: faktoriyel(1000)
```

```
Başlangıç zamanı:      1708696712.063452
40238726007709377354370243392300398571937486421071463254379991042993851239862902059204420848696940480047998861019719605863166687299480855890132382966994459099742450
Bitiş zamanı:   1708696712.065585
Geçen zamanı:   0.002132892608642578
```

```
In [65]: faktoriyel(1558)
```

```
Başlangıç zamanı:      1708696775.3985338
24248584011698758569875493165302612435293363963547008427157499794606461100582336517823926480306255758906032958385491360862036314949043726985781819990572120555357858
```

Bitiş zamanı: 1708696775.403476
Geçen zamanı: 0.004942178726196289

In []: