Python'da gömülü fonksiyonlar

map() fonksiyonu

```
def double(x):
    return x*2

liste = range(0,10,1)

map(double,liste)
```

```
<map at 0x1915821c040>
```

```
list(map(double,liste))
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
import numpy as np

def sin(x):
    return np.sin(np.deg2rad(x))

l = [x for x in range(0,105,15)]

list(map(sin, l))
```

```
[0.0,
 0.25881904510252074,
 0.49999999999999994,
 0.7071067811865476,
 0.8660254037844386,
 0.9659258262890683,
 1.0]
```

```
list(map(lambda x: x **2, (1,2,3,4,5)))
```

```
[1, 4, 9, 16, 25]
```

```
l1 = [1,2,3,4,5]
l2 = [6,7,8,9,10]
l3 = [11,12,13,14,15,16,17]

list(map(lambda x,y : x*y, l1,l2))  # l1 ve l2 listesinin elemanlarını çarpar, skaler çarpım gibi
```

```
[6, 14, 24, 36, 50]
```

```
list(map(lambda x,y,z : x*y*z, l1,l2,l3)) # indiz indiz çarpılır, fazla terimler ihmal edilir.
```

```
[66, 168, 312, 504, 750]
```

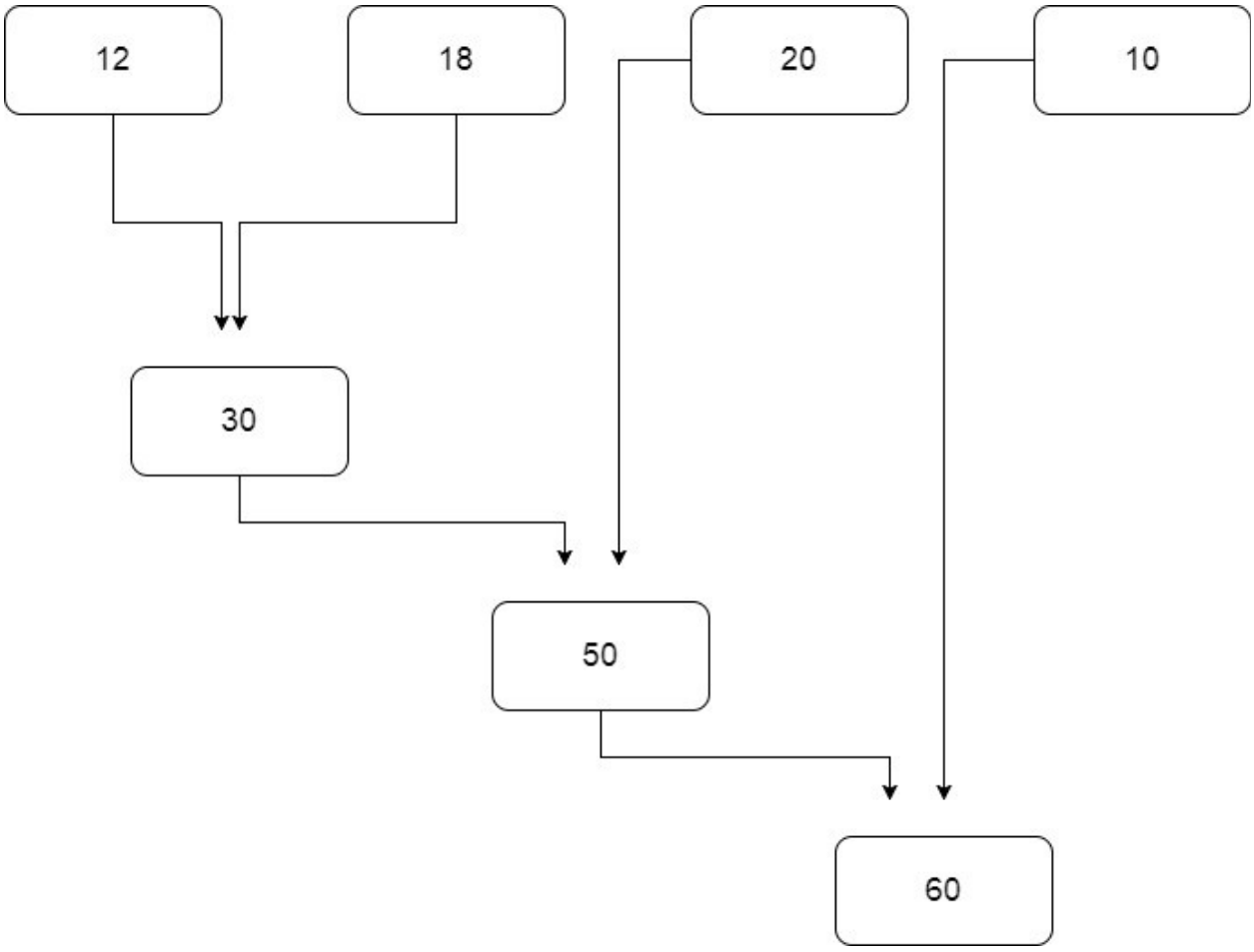
Reduce fonksiyonu

```
from functools import reduce

reduce(lambda x,y : x+y, [12, 18, 20, 10])
```

```
60
```

```
from IPython.display import Image
Image(filename = "reduce.jpeg")
```




```
reduce(lambda x,y: x*y, [1,2,3,4,5])
```

```
120
```

```
reduce(lambda x,y: x+y, [1,2,3,4,5])
```

```
15
```

```
def maksimum(x,y):
    if (x > y):
        return x
    else:
        return y
```

```
maksimum(7,9)
```

```
9
```

```
maksimum(4,2)
```

```
4
```

```
reduce(maksimum, [-2,3,1,4,10]) # listenin maksimum elemanı bulmak
```

```
10
```

filter fonksiyonu

```
list(filter(lambda x : x % 2 == 0, range(1,15)))
```

```
[2, 4, 6, 8, 10, 12, 14]
```

```
def asal(x):
    i = 2
    if(x == 1):
        return False # asal değil
    elif(x == 2):
        return True # asal sayı
    else:
        while (i < x):
            if (x % i == 0):
                return False # Asal değil
            i += 1
        return True
```

```
asal(2)
```

```
True
```

```
asal(6)
```

```
False
```

```
list(filter(asal, range(1,100)))
```

```
[2,
 3,
 5,
 7,
 11,
 13,
 17,
 19,
 23,
 29,
 31,
 37,
 41,
 43,
 47,
 53,
 59,
 61,
 67,
 71,
 73,
 79,
 83,
 89,
 97]
```

```
list(filter(asal, range(1,20)))
```

```
[2, 3, 5, 7, 11, 13, 17, 19]
```

zip()

```
l1 = [1,2,3,4,5]
l2 = [6,7,8,9,10,11,12]

i = 0
sonuç = list()
while (i < len(l1) and i < len(l2)):
    sonuç.append((l1[i],l2[i]))
    i += 1
print(sonuç)
```

```
[(1, 6), (2, 7), (3, 8), (4, 9), (5, 10)]
```

```
list(zip(l1, l2))
```

```
[(1, 6), (2, 7), (3, 8), (4, 9), (5, 10)]
```

```
l1 = [1,2,3,4,5]
l2 = [6,7,8,9,10,11,12]
l3 = ["Python", "C++", "Java", "Reduce"]

list(zip(l1, l2, l3))
```

```
[(1, 6, 'Python'), (2, 7, 'C++'), (3, 8, 'Java'), (4, 9, 'Reduce')]
```

```
for i, j in zip(l1, l2):
    print("i:", i, "\tj:", j)
```

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

```
for i, j, k in zip(l1, l2, l3):
    print("i:", i, "\tj:", j, "\tk:", k)
```

```
i: 1  j: 6  k: Python
i: 2  j: 7  k: C++
i: 3  j: 8  k: Java
i: 4  j: 9  k: Reduce
```

```
s1 = {"elektron": 1, "proton": 2, "nötron": 3}
s2 = {"beta_eksi": 1, "pozitron" : 2, "gama": 4}

list(zip(s1, s2))
```

```
[('elektron', 'beta_eksi'), ('proton', 'pozitron'), ('nötron',
'gama')]
```

```
list(zip(s1.values(),s2.values()))
```

```
[(1, 1), (2, 2), (3, 4)]
```

```
s1.values()
```

```
dict_values([1, 2, 3])
```

enumerate() fonksiyonu

```
liste = ["elektron", "proton", "nötron", "alfa", "beta", "gama"]

sonuç = []

i = 0

for j in liste:
    sonuç.append((i, j))
```

```
    i += 1
print(sonuç)
```

```
[(0, 'elektron'), (1, 'proton'), (2, 'nötron'), (3, 'alfa'), (4,
'beta'), (5, 'gama')]
```

```
list(enumerate(liste))
```

```
[(0, 'elektron'),
 (1, 'proton'),
 (2, 'nötron'),
 (3, 'alfa'),
 (4, 'beta'),
 (5, 'gama')]
```

```
for i,j in enumerate(liste):
    print(i,j)
```

```
0 elektron
1 proton
2 nötron
3 alfa
4 beta
5 gama
```

```
liste = ["a", "b", "c", "d", "e", "f", "g", "h"]

for i, j in enumerate(liste):
    if(i % 2 == 0):
        print("eleman:", j) # listede, indizi çift olan elemanlar
```

```
eleman: a
eleman: c
eleman: e
eleman: g
```

all() ve any() fonksiyonları

```
def h(l):
    for i in l:
        if not i: # i False ise not i True olur ve return ile False
döndürüyoruz
            return False
    return True # bütün değerler True ise True, en az biri False ise
False döndürmek istiyoruz.

liste = [True, True, False, True, True]

h(liste)
```

```
False
```

```
l2 = [True, True, True, True]
```

```
h(l2)
```

```
True
```

```
l2 = [-2, 1, 2, 3, 4]
```

```
h(l2)
```

```
True
```

```
def h2(l):
    for i in l:
        if i:
            return True
    return False # herhangi bir değer True ise True döner
```

```
h2(liste)
```

```
True
```

```
h2(l)
```

```
True
```

```
l3 = [0,0,0,0,0,0]
```

```
h2(l3)
```

```
False
```

```
l4 = [0,0,0,0,0,0,1]
```

```
h2(l4)
```

```
True
```

all() fonksiyonu

```
liste = [True, True, False, True, True]
```

```
all(liste)
```

```
False
```

```
all([True,True,True])
```

```
True
```

any() fonksiyonu

```
any(liste)
```

```
True
```

```
any([False, False, False])
```

```
False
```

```
import pandas as pd

# İki girişli AND kapısının tüm mümkün giriş kombinasyonları
girisler = [(False, False), (False, True), (True, False), (True,
True)]

# Doğruluk tablosunu oluştur
df = pd.DataFrame(girisler, columns=['A', 'B'])

# AND işlemini uygula ve sonucu yeni bir sütun olarak ekle
df['A AND B'] = df.apply(lambda row: row['A'] and row['B'], axis=1)
df['A OR B'] = df.apply(lambda row: row['A'] or row['B'], axis=1)

# all ve any fonksiyonlarını kullanarak kontrol et
# all() fonksiyonu, tüm değerler True ise True döndürür.
# any() fonksiyonu, herhangi bir değer True ise True döndürür.
df['all'] = df.all(axis=1)
df['any'] = df.any(axis=1)

print(df)
```

```
       A      B  A AND B  A OR B    all    any
0  False  False    False   False  False  False
1  False   True    False    True  False   True
2   True  False    False    True  False   True
3   True   True     True    True   True   True
```

Örnek 1

Elinizde bir dikdörtgenin kenarlarını ifade eden sayı çiftlerinin bulunduğu bir liste olsun.

```
    [(3,4),(10,3),(5,6),(1,9)]
```

Şimdi kenar uzunluklarına göre dikdörtgenin alanını hesaplayan bir fonksiyon yazın ve bu listenin her bir elemanına bu fonksiyonu uygulayarak ekrana şöyle bir liste yazdırın.

```
    [12, 30, 30, 9]
```

```
def alan_hesapla(demet):
    return demet[0] * demet[1]

liste = [(3,4),(10,3),(5,6),(1,9)]

print(list(map(alan_hesapla,liste)))
```

```
[12, 30, 30, 9]
```

Örnek 2

Elinizden her bir elemanı 3'lü bir demet olan bir liste olsun.

```
 [(3,4,5),(6,8,10),(3,10,7)]
```

Şimdi kenar uzunluklarına göre bu kenarların bir üçgen olup olmadığını dönen bir fonksiyon yazın ve sadece üçgen belirten kenarları bulunduran listeyi ekrana yazdırın.

```
 [(3, 4, 5), (6, 8, 10)]
```

*** Not: filter() fonksiyonunu kullanmaya çalışın. ***

```
def üçgen_mi(demet):

    if (abs(demet[0]+demet[1]) > demet[2] and abs(demet[0]+demet[2]) >
demet[1] and abs(demet[1]+demet[2]) > demet[0]):
        return True
    else:
        return False

liste = [(3,4,5),(6,8,10),(3,10,7)]

print(list(filter(üçgen_mi,liste)))
```

```
[(3, 4, 5), (6, 8, 10)]
```

Örnek 3

Elinizde şöyle bir liste bulunsun.

```
[1,2,3,4,5,6,7,8,9,10]
```

Bu listenin içindeki çift sayıların toplamını ekrana yazdıran bir fonksiyon yazın.

Not: İlk önce filter() fonksiyonu ile çift sayıları ayıklayın. Daha sonra reduce() fonksiyonunu kullanın.

```
from functools import reduce
liste = [1,2,3,4,5,6,7,8,9,10]

filtre = list(filter(lambda x : x % 2 == 0,liste))

print(reduce(lambda x,y : x + y,filtre))
```

```
30
```

Örnek 4

Elinizde isimlerin ve soyisimlerin bulunduğu iki tane liste olsun.

```
    isimler ----->
["Kerim","Tarık","Ezgi","Kemal","İlkay","Şükran","Merve"]

    soyisimler ----->
["Yılmaz","Öztürk","Dağdeviren","Atatürk","Dikmen","Kaya","Polat"]
```

Bu isimleri ve soyisimleri sırasıyla eşleştirin ve ekrana alt alta isimleri ve soyisimleri yazdırın.

Not: zip() fonksiyonunu kullanmaya çalışın.

```
isimler = ["Kerim","Tarık","Ezgi","Kemal","İlkay","Şükran","Merve"]

soyisimler =
["Yılmaz","Öztürk","Dağdeviren","Atatürk","Dikmen","Kaya","Polat"]

for i,j in zip(isimler,soyisimler):
    print(i, "\t", j)
```

```
Kerim       Yılmaz
Tarık       Öztürk
Ezgi    Dağdeviren
Kemal       Atatürk
İlkay       Dikmen
Şükran      Kaya
Merve       Polat
```