

```
#pythonda matrisi nasıl ifade edeceğiz?
#list yapısını kullanabiliriz.
#iki list iki matris olsun.
#iki tane 4x1 lik matris oluşturalım
v_1=[1,2,3,4].
v_2=[12,22,13,47].

#simdi bunları toplayabiln fonksiyonu oluşturalım
def my_f_1(v1,v2):
    k=len(v1)
    v3=[0]*k
    for i in range(k):
        v3[i]=v1[i]+v2[i]
    return v3
```

```
my_f_1(v_1,v_2)
```

```
#daha farklı da su şekilde yazabiliriz
def add_vec(v1,v2):
    return [x+y for (x,y) in zip(v1,v2)].
add_vec(v_1,v_2)
```

```
#simdi de bir vektörü skalerle çarpmak istiyoruz. Fonksiyonu su şekilde:
def multiply_scalar(v1,alpha):
    return [x*alpha for x in v1].
multiply_scalar(v_1,2)
```

```
#şimdi de iki vektörün farklı skalerlerle çarpılıp toplanmasına
#bakalım

def my_f_5(v1,alpha,v2,beta):

    v1_1=multiply_scalar(v1,alpha)
    v2_1=multiply_scalar(v2,beta)

    return add_vec(v1_1,v2_1)
v_1=[1,2,3,4].
v_2=[12,22,13,47].

my_f_5(v_1,2,v_2,2) #üstte tanımlanmış her iki vektörü de iki ile çarptık
#ve birbirleriyle topladık
```

```
#vektör tanımlamak başka bir yöntemse dict oluşturmak.
#sözlük, anahtar-değer olarak ikiye ayrılır. Tıpkı sözlük gibi anahtar ve
#anahtara karşılık gelen değer/değerler listesi gibi düşünülebilir
#şimdi örnek bir dict oluşturalım

h={1:['1',2,'3'], 2:['1', 22, '33']}
h
```

```
h.keys() #anahtarları veriyor
```

```
for key in h.keys():
    print(h[key]) #verilen key anahtarına karşılık gelen değeri yazıyor
```

```
#simdi de dict yapısıyla bir vektör oluşturmayı deneyelim
v3={1:10,2:20,3:30,4:40}
v3
```

```
for key in v3.keys():
    print(v3[key])
```

```
d1={'a','b','c','d'} #küme oluşturduk ve elemanlarını yerleştirdik
type(d1)
```

```
f1={'a':1,'b':2} #f1 ise bir dictionary oldu
#simdi bu iki yapıyı birleştirip tuple yapalım

v10=(d1,f1)
v10
```

```
v10={'D':d1, 'F':f1} #v10 da simdi dict oldu.
v10, v10.keys()
```

```
#simdi v3'ü güncelleyelim dict olarak
v3={15:10, 29:20, 3000:30,4000:40}
```

```
class Vector:
#sınıf tanımladık
    def __init__(self , labels, function):
        self.D=labels
        self.f=function
```

```
labels={'A', 'B','C'}
function={'A':1}
v=Vector(labels,function)
v
```

```
for d in v.D:
    if d in v.f:
        print(v.f[d])
```

```
def create_zero_vector(D): #0 vektör oluşturduk
    f={}
    for d in D:
        f[d]=0
    return Vector(D,f)
```

```
v=create_zero_vector({'a', 'b', 'c', 'd'})
#küme olduğu için bizim verdiğimiz sırada oluşturmuyor elemanları
```

```
for d in v.D:  
    if d in v.f:  
        print(d, v.f[d])
```

```
#simdi de belirli bir elemanı ekleyelim, değıştirelim  
def setitem(V,d,val):  
    V.f[d]=val  
setitem(v,'c',10)  
#tekrar for döngüsünü uygularsak c değerini 10 yapmış olacak
```