

PYTHON PROGRAMLAMA DİLİ

enumerate, zip, for-if

Dr. Öğr. Üyesi M. Ozan AKI

ENUMERATE

String, list, tuple gibi eleman dizileri tutan yapılarda, her bir elemana indis numarası verir.

Bir **string** ya da **list** elemanlarının indis numaraları ile ilgili işlem yapmak gerektiğinde

enumerate()

Fonksiyonu ile indis numaraları atanır.

Her bir eleman, indis numarası ve elemanın kendinden oluşan bir **tuple** olarak döner

```
>>> s='Merhaba'
>>> for e in enumerate(s):
...     print(e)
...
...
...
(0, 'M')
(1, 'e')
(2, 'r')
(3, 'h')
(4, 'a')
(5, 'b')
(6, 'a')
>>> |
```

ZIP

zip, iki listenin elemanlarını, aynı indisteki elemanları karşılıklı olarak bire-bir eşleştirerek, her bir elemanı bir **tuple** olacak şekilde yeni bir liste döndürür.

Diğer bir deyişle, iki listeyi birleştirir ve tek bir liste haline getirir.

```
>>> listel=[1,2,3,4,5]
>>> liste2=['a','b','c','d','e']
>>>
>>> liste=list(zip(listel,liste2))
>>> print(liste)
[(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd'), (5, 'e')]
>>> |
```

LIST COMPREHENSION

range nesnesi ile istenen düzende artan ya da azalan sayı dizileri üretebiliyorduk.

Ancak başka bir **range** ya da başka bir listeden, her bir eleman üzerinde çeşitli işlemler yaparak yeni bir liste oluşturmak için köşeli parantez içinde **for** ifadesi kullanılır.

Bu yöntem ile, bir **range** üzerinden ya da başka listelerden daha karmaşık yapıda yeni bir liste elde etmek için kullanılır.

```
>>> liste=['A' for i in range(5)]
>>> print(liste)
['A', 'A', 'A', 'A', 'A']
>>>
>>> sifirlar=[0 for x in range(10)]
>>> print(sifirlar)
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
>>>
```

LIST COMPREHENSION

for ifadesi, mevcut listedeki her bir eleman için döngü değişkenine elemanı aktarır. **for** önüne yazılan ifade ile, döngü değişkeni üzerinde işlem yapılır ve yeni listenin her bir elemanı hesaplanan sonuç olur.

yeni_liste = [ifade for item in başka_liste]

```
>>> kareler=[x**2 for x in range(5)]
>>> print(kareler)
[0, 1, 4, 9, 16]
>>>
>>> kupler=[x**3 for x in range(5)]
>>> print(kupler)
[0, 1, 8, 27, 64]
>>> |
```

LIST COMPREHENSION

Eğer verilen listenin her bir elemanı değil de sadece bazı elemanları üzerinde işlem yapmak istersek, ***başka_liste*** ifadesinden sonra bir **if** koşulu yazılabilir.

Bu şekilde sadece koşulu sağlayan elemanlar işleme sokulur.

yeni_liste =[ifade for item in başka_liste if kosul == True]

```
>>> tek_kareler=[sayi**2 for sayi in range(10) if sayi%2!=0]
>>> print(tek_kareler)
[1, 9, 25, 49, 81]
>>>
>>> cift_kupler=[sayi**3 for sayi in range(10) if sayi%2==0]
>>> print(cift_kupler)
[0, 8, 64, 216, 512]
>>>
```