

Sayı Operasyonları

Aşağıdaki hücrelerde, şu işlemleri print komutunu kullanarak ekrana yazdır:

```
In [1]: # 3 ve 2'yi toplamı için bu alanı kullanabilirsin:  
print(3+2)
```

5

```
In [1]: # 5 ve 4'ü çarpımı için bu alanı kullanabilirsin:  
print(5*4)
```

20

```
In [2]: # 5 ve 4'ün bölümünden kalan için bu alanı kullanabilirsin:  
print(5%4)
```

1

Aşağıdaki hücreleri kullanarak, şu işlemleri yanıtlandır:

```
In [4]: # 2.3 ve 4.7'nin toplama işleminin neticesi hangi veri tipindedir?  
float
```

Out [4]: float

```
In [5]: # 2.3 ve 4.7'nin toplama işleminin veri tipini, komut kullanarak ekrana bastır?  
print(type(2.3 + 4.7))
```

<class 'float'>

Yazılar

```
In [7]: strvar = 'Yakın Kampüs'
```

Yukarıdaki değişkeni kullanarak, aşağıdaki hücrelerde, şu işlemleri yanıtlandır:

```
In [10]: # "ü" harfini index komutu kullanmadan döndür (İpucu: [])  
strvar[10]
```

Out [10]: 'ü'

```
In [11]: # "ü" harfinin konumunu, index komutu kullanarak döndür  
strvar.index('ü')
```

Out [11]: 10

```
In [12]: # strvar.upper().lower() ifadesinin cevabi nedir?
```

```
strvar.upper().lower()
```

Out [12]: 'yakın kampüs'

```
In [13]: # split() komutunu boşluk(space) dışında başka bir parametre ile kullan.  
strvar.split('a')
```

Out [13]: ['Y', 'kın K', 'mpüs']

Boolean

```
In [15]: a = True  
b = False  
c = 'True'
```

Yukarıdaki değişkenleri kullanarak, aşağıdaki hücrelerde, şu işlemleri yanıtlandır:

```
In [16]: # a == b sorgusu cevap olarak ne döndürür?  
a == b
```

Out [16]: False

```
In [17]: # a == c sorgusu cevap olarak ne döndürür?  
a == c
```

Out [17]: False

```
In [18]: # a!=b sorgusu cevap olarak ne döndürür?  
a!=b
```

Out [18]: True

```
In [4]: # ! ve not hangi senaryoda birbirinin yerine kullanılamaz? (İpucu: >)  
# > ya da < isareti ile ! aynı anda kullanılamaz, ama not ifadesi kullanılabilir
```

```
In [19]: # type(3) == type('3') sorgusunu cevap olarak ne döndürür?  
type(3) == type('3')
```

Out [19]: False

Listeler ve Setler

```
In [20]: liste = [1, 'a', 2, 3, True, 4, 5, 'True', '1']
```

Yukarıdaki değişkeni kullanarak, aşağıdaki hücrelerde, şu işlemleri yanıtlandır:

```
In [21]: # Listenin son elemanını nasıl bulabiliriz?  
liste[-1]
```

Out [21]: '1'

```
In [24]: # Listenin 2. ve 4. elemanlarını içeren yeni bir liste oluştur (İpucu[::])  
liste[2:6:3]
```

Out [24]: [2, 4]

```
In [32]: # Listeyi metod kullanarak nasıl ters sıralarsınız?  
liste.reverse()  
print(liste)
```

['1', 'True', 5, 4, True, 3, 2, 'a', 1]

```
In [33]: # Listeyi [::] kullanarak nasıl ters sıralarsınız?  
liste = [1, 'a', 2, 3, True, 4, 5, 'True', '1']  
liste[::-1]
```

Out [33]: ['1', 'True', 5, 4, True, 3, 2, 'a', 1]

```
In [40]: # Eğer yukarıdaki listeyi, set'e çevirecek olsak eleman sayısı farklı olur muydu? Ne  
# olmazdı çünkü her eleman 1 defa tekrar ediyor, ama True'yu teke indiriyor!  
len(liste) == len(set(liste))
```

Out [40]: False

```
In [42]: ic_ice_liste = [1,2,3,[4,5]]
```

Yukarıdaki değişkeni kullanarak, aşağıdaki hücrelerde, şu işlemleri yanıtlandır:

```
In [45]: # 5 değerine nasıl ulaşırsınız?  
print(ic_ice_liste[-1][-1])  
print(ic_ice_liste[3][1])  
print(ic_ice_liste[3][-1])  
print(ic_ice_liste[-1][1])
```

5
5
5
5

```
In [46]: # ic_ice_liste değişkeninin son konumunda bulunan elemanını listeden atın ve bu kıs  
artik = ic_ice_liste.pop()  
artik
```

Out [46]: [4, 5]

```
In [48]: # pop komutunu kullanmayarak listeyi nasıl [1,2,[4,5]] şekline çevirebilirsiniz? (H  
ic_ice_liste = [1,2,3,[4,5]]  
yeni_liste = ic_ice_liste[:2] + [ic_ice_liste[-1]]  
yeni_liste
```

Out [48]: [1, 2, [4, 5]]

Dictionary

```
In [ ]: # Dictionary veri tipinin listeden farkı nedir?  
# dictionary'de sıranın bir önemi yoktur, çok boyutlu elemanlardır, liste koleksiyon
```

```
In [ ]: # Dictionary oluştururken kullanılan anahtar ve değer çiftinin Python'daki karşılığı
# key ve value
```

```
In [49]: my_dict = {'isim': 'Mesut', 'yas':32, 'lokasyon': {'yasadigi':'Berlin', 'dogdugu':
```

```
In [50]: # 32 değerine nasıl ulaşırsınız?
print(my_dict['yas'])
print(my_dict.get('yas'))
```

```
32
32
```

```
In [51]: # isim anahtarına karşılık gelen değeri, kendi isminizle değiştirin:
my_dict['isim'] = 'Oguz'
my_dict
```

```
Out [51]: {'isim': 'Oguz',
' yas': 32,
'lokasyon': {'yasadigi': 'Berlin', 'dogdugu': 'Istanbul'}}
```

```
In [52]: # my_dict değişkenindeki 'Istanbul' değerine nasıl ulaşabiliriz?
print(my_dict['lokasyon']['dogdugu'])
print(my_dict.get('lokasyon').get('dogdugu'))
```

```
Istanbul
Istanbul
```

```
In [53]: # my_dict değişkenindeki bütün anahtar değerlerine nasıl ulaşırız?
my_dict.keys()
```

```
Out [53]: dict_keys(['isim', 'yas', 'lokasyon'])
```

Tuple

```
In [ ]: # Tuple ile liste arasındaki farklar nelerdir?
# tuple immutable yani degisime acik degildir
```

Genel Konular

```
In [ ]: # print komutu ile ve print komutsuz yazdırmanın farkı nedir? Her durumda birbiri ye
# print ile istedigimiz kadar satir yazdirabiliriz, print kullanilmazsa sadece son s
```

```
In [ ]: # String ifadelerinin içerisinde \t ile \n kullanmanın farkı nedir?
# t tab ifadesidir, bir kac satir bosluk birakir, n ifadesi yeni satira gecirtir
```

```
In [ ]: # değişken isimlendirmelerinde dikkat edilmesi gereken hususlar nelerdir?
# bosluk, rezerve edilmiş karakterler kullanilamaz, degisken ismi sayi ile baslayama
```

In []:

```
# [] {} ve () işaretlerini veri tipleri ile eşleştirin:  
# '' -> String  
# [] -> liste  
# {} -> dictionary, set  
# () -> tuple
```