



PYTHON PROGRAMLAMA DİLİ

VERİ TİPLERİ
dictionary

Dr. Öğr. Üyesi M. Ozan AKI

VERİ TİPLERİ

Python dilinde, diğer programlama dillerinde olduğu gibi farklı veri tipleri bulunmaktadır.

int, **float** ve **str** sadece bir değişken tanımlarken,

List, **Tuple**, **Dictionary** ve **Set** veri tipleri, çok sayıda veriyi sistematik bir şekilde saklamak ve işlemek için kullanılmaktadır.

Veri Tipi	Açıklama
int	Tam sayı 123
float	Ondalıklı sayılar 12.34
str	Karakter dizileri '...' “...”
List	Listeler [...]
Tuple	Demetler (...)
Dictionary	Sözlük {...}
Set	Küme {...}

DICTIONARY (SÖZLÜK)

Dictionary (Sözlük), verileri liste ve demetlerden farklı olarak indise bağlı değil, anahtarlara bağlı olarak saklar.

Anahtar:Değer (**Key:Value**)

Çifti şeklinde tutulan bu veriler {...} süslü parantez içerisinde tanımlanır.

Anahtar ve Değer çifti arasına : (iki nokta üst üste) karakteri konur.

Sözlük elemanlarına Anahtar (Key) elemanı ile erişilir.

Her Anahtardan (Key) sadece bir tane bulunabilir.

```
>>> sozluk={'bir':'one','iki':'two','uc':'three'}
>>> print(sozluk)
{'bir': 'one', 'iki': 'two', 'uc': 'three'}
>>> len(sozluk)
3
>>> sozluk['dort']='four'
>>> print(sozluk)
{'bir': 'one', 'iki': 'two', 'uc': 'three', 'dort': 'four'}
>>> len(sozluk)
4
>>> del sozluk['iki']
>>> print(sozluk)
{'bir': 'one', 'uc': 'three', 'dort': 'four'}
>>> len(sozluk)
3
>>> |
```

DICTIONARY (SÖZLÜK)

Sözlüklerin, elemanları üzerinde çeşitli işlemler yapan “Metotları” vardır.

Bu metotlara, liste adından sonra bir . (nokta) konulduktan sonra erişilir.

Metotlar parantez içeirisinde parameter alabilirler.

Metotlar, bir değişkene atayabileceğiniz bir değer döndürebilirler.

```
dictionary.items()  
dictionary.keys()  
dictionary.values()  
dictionary.get()  
dictionary.setdefault()  
dictionary.pop()  
dictionary.popitem()  
dictionary.update()  
dictionary.fromkeys()  
dictionary.copy()
```

DICTIONARY (SÖZLÜK)

d.items(): Sözlükteki anahtar-değer (key-value) çiftlerini demet (tuple) olarak döndürür.

d.keys(): Sözlükteki sadece anahtarları (key) demet olarak döndürür.

d.values(): Sözlükteki sadece değerleri (value) demet olarak döndürür.

```
>>> sozluk.items()
dict_items([('bir', 'one'), ('uc', 'three'), ('dort', 'four')])
>>> sozluk.keys()
dict_keys(['bir', 'uc', 'dort'])
>>> sozluk.values()
dict_values(['one', 'three', 'four'])
>>> sozluk.get('uc')
'three'
>>> sozluk.get('dokuz')
None
>>> #None doner
>>> sozluk.get('dokuz', 'nine')
'nine'
>>>
>>> print(sozluk)
{'bir': 'one', 'uc': 'three', 'dort': 'four'}
```

d.get(k, m): Eğer k anahtar olarak sözlükte varsa, ilişkili değeri döndürür. Yoksa verilen m değerini döndürür (bu değer verilmediyse None döndürür).

DICTIONARY (SÖZLÜK)

d.setdefault(k, m): Eğer k anahtar olarak sözlükte varsa, ilişkili değeri döndürür. Yoksa yeni bir eleman olarak ekler ve m parametresini değer olarak kullanır (verilmediyse None kullanır).

d.pop(k, m): 'get' ile aynı işi yapar ama sözlükten ilgili elemanı da siler. Eğer k sözlükte anahtar olarak yoksa KeyError hatası verir (del komutu da aynı)

d.popitem(): Kümelerdeki pop gibi keyfi bir eleman siler ve bu elemanı döndürür.

```
>>> sozluk={'bir':'one','iki':'two','uc':'three'}
>>> sozluk.setdefault('iki')
'two'
>>> sozluk.setdefault('dort','four')
'four'
>>> print(sozluk)
{'bir': 'one', 'iki': 'two', 'uc': 'three', 'dort': 'four'}
>>> sozluk.pop('iki')
'two'
>>> print(sozluk)
{'bir': 'one', 'uc': 'three', 'dort': 'four'}
>>> sozluk.popitem()
('dort', 'four')
>>> print(sozluk)
{'bir': 'one', 'uc': 'three'}
>>> |
```

DICTIONARY (SÖZLÜK)

d.update(s): s sözlüğündeki anahtarlarla aynı olan d sözlüğünde anahtarlar varsa d sözlüğündeki değerleri s sözlüğündekiler ile günceller. Bulunamayan anahtarlar için anahtar-değer ikililerini de d sözlüğüne ekler. s bir sözlük yerine anahtar-değer çiftlerini içeren bir iterable olabilir. s verilmezse d aynı kalır.

d.fromkeys(s, m): içine aldığı s iterable'ı ile anahtarları oluşturarak yeni bir sözlük döndürür. Seçimlik olan m ile her anahtar için sabit bir başlangıç değeri verilebilir (verilmezse None kullanır)

```
>>> sozluk={'marka':'opel','renk':'mavi','model':2018,'motor':1.2}
>>> print(sozluk)
{'marka': 'opel', 'renk': 'mavi', 'model': 2018, 'motor': 1.2}
>>> sozluk.update({'model':2021,'motor':1.5,'vites':'otomatik'})
>>> print(sozluk)
{'marka': 'opel', 'renk': 'mavi', 'model': 2021, 'motor': 1.5, 'vites': 'otomatik'}
>>> yeni_sozluk=sozluk.fromkeys(['marka','model','renk'],'bos')
>>> print(yeni_sozluk)
{'marka': 'bos', 'model': 'bos', 'renk': 'bos'}
>>> |
```