```
list1 = []
print(type(list1))
```

```
<class 'list'>
```

```
list2 = [10, 30, 60]
list3 = [10.77, 30.66, 60.89]
list4 = ['one', 'two', 'three']
list5 = ['Doğu', 25, [50, 100], [150, 90]]
list6 = [100, 'Doğu', 17.765]
list7 = ['Doğu', 25, [50, 100], [150, 90], {'John', 'David'}]
len(list6)
```

```
3
```

```
#Dizin İşlemleri (Indexing)
list2[0]
```

```
10
```

```
list4[0]
```

```
'one'
```

```
list4[0][0]
```

```
'o'
```

```
list4[-1]
```

```
'three'
```

```
list5[-1]
```

```
[150, 90]
```

```
#Dilimleme (Slicing)
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight']
mylist[0:3]
```

```
['one', 'two', 'three']
```

```
mylist[2:5]
```

```
['three', 'four', 'five']

mylist[:3]

['one', 'two', 'three']

mylist[-3:]

['six', 'seven', 'eight']

mylist[:]

['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

#Add, Remove, Change

mylist

['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

mylist.append('nine')
mylist

['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight',
'nine']

mylist.insert(9, 'ten')
mylist

['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight',
'nine', 'ten']

mylist.insert(1, 'ONE')

mylist

['one',
 'ONE',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten']

mylist.remove('ONE')
mylist

['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight',
'nine', 'ten']
```

```
mylist.pop()
mylist
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight',
'nine']
```

```
mylist.pop(8)
mylist
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
del mylist[7]
mylist
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

```
[1, 2, 3, 'four', 'five', 'six', 'seven']
```

```
mylist.clear()
mylist
```

```
[]
```

```
del mylist
mylist
```

```
---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
<ipython-input-35-e3998fd6e4af> in <module>
      1 del mylist
----> 2 mylist

NameError: name 'mylist' is not defined
```

```
#Kopyalama

mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight']

mylist1 = mylist

id(mylist), id(mylist1)
```

```
(2362670482496, 2362670482496)
```

```
mylist2 = mylist.copy()
```

```
id(mylist2)
```

```
2362670300992
```

```
mylist[0] = 1
```

```
mylist
```

```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
mylist1
```

```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
mylist2
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
#Join (Birleştirmek)
```

```
list1 = ['one', 'two', 'three', 'four']
list2 = ['five', 'six', 'seven', 'eight']
```

```
list3 = list1 + list2
list3
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
list1.extend(list2)
list1
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
#Üyelik (Membership)
```

```
list1
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
'one' in list1
```

```
True
```

```
'ten' in list1
```

```
False
```

```
if 'three' in list1:
    print("Three is present in the list")
else:
    print("Three is not present in the list")
```

```
Three is present in the list
```

```
if 'eleven' in list1:
    print("Eleven is present in the list")
else:
    print("Eleven is not present in the list")
```

```
Eleven is not present in the list
```

```
#Reverse-Sort (Ters Çevirme-Sıralama)
```

```
list1
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
list1.reverse()
```

```
list1
```

```
['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

```
list1 = list1[::-1]
list1
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
mylist3 = [9, 5, 2, 99, 12, 88, 34]
mylist3.sort() #ascending (küçükten büyüğe)
mylist3
```

```
[2, 5, 9, 12, 34, 88, 99]
```

```
mylist3 = [9, 5, 2, 99, 12, 88, 34]
mylist3.sort(reverse=True)
mylist3
```

```
[99, 88, 34, 12, 9, 5, 2]
```

```
mylist4 = [88, 65, 33, 21, 11, 98]
sorted(mylist4)
```

```
[11, 21, 33, 65, 88, 98]
```

```
mylist4
```

```
[88, 65, 33, 21, 11, 98]
```

```
list1
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
for i in list1:
    print(i)
```

```
one
two
```

```
three
four
five
six
seven
eight
```

```
for i in enumerate(list1):
    print(i)
```

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
(6, 'seven')
(7, 'eight')
```

```
#Sayma (Count)
list10 = ['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

```
list10.count('one')
```

```
3
```

```
list10.count('two')
```

```
2
```

```
list10.count('four')
```

```
1
```

```
#All -> Listedeki bütün elemanlar True ise True üretir, herhangi biri
False ise False üretir.
#Any -> Listedeki herhangi bir eleman True ise True üretir, hiç True
yoksa False üretir.
```

```
L1 = [1,2,3,4,0]
```

```
all(L1)
```

```
False
```

```
any(L1)
```

```
True
```

```
L2 = [1,2,3,4,True,False]
```

```
all(L2)
```

```
False
```

```
any(L2)
```

```
True
```

```
L3 = [1,2,3,True]
```

```
all(L3)
```

```
True
```

```
any(L3)
```

```
True
```

```
#List Comprehensions
```

```
mystring = "WELCOME"
mylist = [i for i in mystring]
mylist
```

```
['W', 'E', 'L', 'C', 'O', 'M', 'E']
```

```
mylist1 = [i for i in range(40) if i % 2 == 0]
```

```
mylist1
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34,
36, 38]
```

```
mylist2 = [i for i in range(40) if i % 2 == 1]
mylist2
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35,
37, 39]
```

```
mylist3 = [num**2 for num in range(10)]
mylist3
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
list1 = [2,3,4,5,6,7,8]
list1 = [i*10 for i in list1]
list1
```

```
[20, 30, 40, 50, 60, 70, 80]
```

```
mylist4 = [i for i in range(200) if i % 3 == 0 if i % 9 == 0 if i % 12
== 0]
mylist4
```

```
[0, 36, 72, 108, 144, 180]
```

```
mystr = "One 1 two 2 three 3 four 4 five 5 six 6789"
numbers = [i for i in mystr if i.isdigit()]
numbers
```

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

```
mystr = "One 1 two 2 three 3 four 4 five 5 six 6789"
letters = [i for i in mystr if i.isalpha()]
letters
```

```
['O',
 'n',
 'e',
 't',
 'w',
 'o',
 't',
 'h',
 'r',
 'e',
 'e',
 'f',
 'o',
 'u',
 'r',
 'f',
 'i',
 'v',
 'e',
 's',
 'i',
 'x']
```

```
#Tuples (Demetler)

tup1 = ()

tup2 = (10, 30, 60)

tup3 = (10.77, 30.66, 60.89)

tup4 = ('one', 'two', 'three')

tup5 = ('Doğu', 26, (50, 100), (150, 90))

tup6 = (100, 'Doğu', 17.765)

tup7 = ('Doğu', 26, [50, 100], [150, 90], {'John', 'David'}, (99, 22,
33))

len(tup7)
```

```
6
```

```
tup2[0]
```

```
10
```

```
tup4[0]
```

```
'one'
```

```
tup4[0][0]
```

```
'o'
```

```
tup4[-1]
```

```
'three'
```

```
tup5[-1]
```

```
(150, 90)
```

```
mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight')
```

```
mytuple[0:3]
```

```
('one', 'two', 'three')
```

```
mytuple[-3:]
```

```
('six', 'seven', 'eight')
```

```
mytuple[-1]
```

```
'eight'
```

```
mytuple[:]
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
#Remove, Change
```

```
del mytuple[0]
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
<ipython-input-113-667a276aa503> in <module>
----> 1 del mytuple[0]

TypeError: 'tuple' object doesn't support item deletion
```

```
mytuple[0] = 1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-114-4cf492702bfd> in <module>
----> 1 mytuple[0] = 1

TypeError: 'tuple' object does not support item assignment
```

```
del mytuple

mytuple
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-116-c6c21778968d> in <module>
----> 1 mytuple

NameError: name 'mytuple' is not defined
```

```
mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight')
mytuple
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
for i in mytuple:
    print(i)
```

```
one
two
three
four
five
six
seven
eight
```

```
for i in enumerate(mytuple):
    print(i)
```

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
```

```
(6, 'seven')
(7, 'eight')
```

```
#Sayma (Count)
mytuple1 = ['one', 'two', 'three', 'four', 'one', 'one', 'two',
'three']
```

```
mytuple1.count('one')
```

```
3
```

```
#Üyelik (Memebership)
```

```
'one' in mytuple1
```

```
True
```

```
'ten' in mytuple1
```

```
False
```

```
if 'eleven' in mytuple1:
    print("Eleven is present in the tuple")
else:
    print("Eleven is not present in the tuple")
```

```
Eleven is not present in the tuple
```

```
#Index Position
```

```
mytuple
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
mytuple.index('one')
```

```
0
```

```
mytuple.index('five')
```

```
4
```

```
#Sorting
```

```
mytuple2 = (43, 67, 99, 12, 6, 90, 67)
```

```
sorted(mytuple2) #ascending
```

```
[6, 12, 43, 67, 67, 90, 99]
```

```
sorted(mytuple2, reverse = True) #descending
```

```
[99, 90, 67, 67, 43, 12, 6]
```

```
#Arama uygulaması
```

```
def main():

    prod_nums = ['V475', 'F987', 'Q143', 'R688']

    search = input("Enter a product number: ")

    if search in prod_nums:
        print(search, 'was found in the list')
    else:
        print(search, 'was not found in the list')

main()
```

```
Enter a product number: A123
A123 was not found in the list
```

```
#Listeye eleman ekleme uygulaması

def main():

    name_list = []

    again = 'y'

    while again == 'y':
        name = input("Enter a name: ")

        name_list.append(name)

        #Eğer başka bir eleman eklenmek isteniyorsa
        print('Do you want to add another name?')
        again = input('y = yes, anything else = no: ')
        print()

        print("Here are the names you entered.")

        for name in name_list:
            print(name)

main()
```

```
Enter a name: Ahmet
Do you want to add another name?
y = yes, anything else = no: y

Here are the names you entered.
Ahmet
Enter a name: Mehmet
Do you want to add another name?
y = yes, anything else = no: y
```

```
Here are the names you entered.
Ahmet
Mehmet
Enter a name: Fatma
Do you want to add another name?
y = yes, anything else = no: y

Here are the names you entered.
Ahmet
Mehmet
Fatma
Enter a name: Ayşe
Do you want to add another name?
y = yes, anything else = no: y

Here are the names you entered.
Ahmet
Mehmet
Fatma
Ayşe
Enter a name: Mustafa
Do you want to add another name?
y = yes, anything else = no: ğ

Here are the names you entered.
Ahmet
Mehmet
Fatma
Ayşe
Mustafa
```

```python
#İki boyutlu bir listeye random (rastgele) eleman ekleme
import random

ROWS = 3
COLS = 4

def main():

    values = [[0, 0, 0, 0],
              [0, 0, 0, 0],
              [0, 0, 0, 0]]

    for r in range(ROWS):
        for c in range(COLS):
            values[r][c] = random.randint(1, 100)

    print(values)

main()
```

```
[[88, 94, 55, 96], [76, 13, 84, 49], [8, 64, 92, 73]]
```

```
#String Metotları
```

```
str1 = "HELLO PYTHON"
```

```
str1
```

```
'HELLO PYTHON'
```

```
str1[len(str1) - 1]
```

```
'N'
```

```
str1[5]
```

```
' '
```

```
str1[-4:]
```

```
'THON'
```

```
str1[:6]
```

```
'HELLO '
```

```
#Update, Delete
```

```
str1
```

```
'HELLO PYTHON'
```

```
str1[0:5] = 'HOLAA'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-157-eb59ceb4fbd5> in <module>
----> 1 str1[0:5] = 'HOLAA'

TypeError: 'str' object does not support item assignment
```

```
del str1
print(str1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-158-6f25af838a38> in <module>
      1 del str1
----> 2 print(str1)
```

```
NameError: name 'str1' is not defined
```

```
s1 = 'Doğu '
s2 = 'Sırt'
s3 = s1 + s2
s3
```

```
'Doğu Sırt'
```

```
mystr1 = 'Hello Everyone'
```

```
for i in mystr1:
    print(i)
```

```
H
e
l
l
o

E
v
e
r
y
o
n
e
```

```
for i in enumerate(mystr1):
    print(i)
```

```
(0, 'H')
(1, 'e')
(2, 'l')
(3, 'l')
(4, 'o')
(5, ' ')
(6, 'E')
(7, 'v')
(8, 'e')
(9, 'r')
(10, 'y')
(11, 'o')
(12, 'n')
(13, 'e')
```

```
list(enumerate(mystr1))
```

```
[(0, 'H'),
 (1, 'e'),
 (2, 'l'),
 (3, 'l'),
 (4, 'o'),
 (5, ' '),
 (6, 'E'),
 (7, 'v'),
 (8, 'e'),
 (9, 'r'),
 (10, 'y'),
 (11, 'o'),
 (12, 'n'),
 (13, 'e')]
```

```
#Membership (Üyelik)
mystr1 = 'Hello Everyone'

print('Hello' in mystr1)
print('Everyone' in mystr1)
print('Hi' in mystr1)
```

```
True
True
False
```

```
#String Partitioning (Bölümleme)

str5 = "Natural Language Processing with Python and R and Java"

L = str5.partition("and")

print(L)
```

```
('Natural Language Processing with Python ', 'and', ' R and Java')
```

```
str5 = "Natural Language Processing with Python and R and Java"

L = str5.rpartition("and")

print(L)
```

```
('Natural Language Processing with Python and R ', 'and', ' Java')
```

```
mystr2 = "   Hello Everyone   "
mystr2
```

```
'   Hello Everyone   '
```

```
mystr2.strip()
```

```
'Hello Everyone'
```

```
mystr2.rstrip()
```

```
'   Hello Everyone'
```

```
mystr2.lstrip()
```

```
'Hello Everyone   '
```

```
mystr2 = "**************Hello Everyone***************All The
Best******************"
```

```
mystr2.strip('*')
```

```
'Hello Everyone***************All The Best'
```

```
mystr2.rstrip('*')
```

```
'**************Hello Everyone***************All The Best'
```

```
mystr2.lstrip('*')
```

```
'Hello Everyone***************All The Best******************'
```

```
mystr3 = 'Hello Everyone***************All The Best'

temizlenmis_mystr3 = ""

for karakter in mystr3:
    if karakter != '*':
        temizlenmis_mystr3 += karakter

print(temizlenmis_mystr3)
```

```
Hello EveryoneAll The Best
```

```
mystr3 = 'Hello Everyone***************All The Best'

start_dizin = mystr3.find('*')
end_dizin = mystr3.rfind('*')
temizlenmis_mystr3 = mystr3[:start_dizin] + ' ' + mystr3[end_dizin +
1:]

print(temizlenmis_mystr3)
```

```
Hello Everyone All The Best
```

```
mystr5 = "   HELLO EVERYONE   "
mystr5.lower()
```

```
'   hello everyone   '
```

```
mystr5.upper()
```

```
'   HELLO EVERYONE   '
```

```
mystr5.replace('HE', 'HO')
```

```
'   HOLLO EVERYONE   '
```

```
mystr5.replace(" ", "")
```

```
'HELLOEVERYONE'
```

```
mystr6 = "one two three one two two three"
mystr6.count("one")
```

```
2
```

```
mystr6.startswith("one")
```

```
True
```

```
mystr6.endswith("three")
```

```
True
```

```
mystr7 = 'one two three four one two two three five five six seven six
seven one'
```

```
mylist = mystr7.split()
mylist
```

```
['one',
 'two',
 'three',
 'four',
 'one',
 'two',
 'two',
 'three',
 'five',
 'five',
 'six',
 'seven',
 'six',
 'seven',
 'one']
```

```
item1 = 40
item2 = 55
item3 = 77

result = "Cost of item1, item2 and item3 are {}, {} and {}"

print(result.format(item1, item2, item3))
```

```
Cost of item1, item2 and item3 are 40, 55 and 77
```

```
#Döküman Oluşturma

str2 = " WELCOME EVERYONE "
str2 = str2.center(100)
print(str2)
```

```
                                                 WELCOME EVERYONE 
```

```
str2 = " WELCOME EVERYONE "
str2 = str2.center(100, '*')
print(str2)
```

```
****************************************** WELCOME EVERYONE ******************************************
```

```
str2 = " WELCOME EVERYONE "
str2 = str2.rjust(50)
print(str2)
```

```
                                   WELCOME EVERYONE 
```

```
str2 = " WELCOME EVERYONE "
str2 = str2.rjust(50, '*')
print(str2)
```

```
****************************** WELCOME EVERYONE 
```

```
str4 = 'one two three four five six seven'
loc = str4.find('five')
print(loc)
```

```
19
```

```
mystr6 = 'abc12309'
print(mystr6.isalpha()) #Sadece karakter
print(mystr6.isalnum()) #Karakter-sayı karışık
print(mystr6.isdecimal()) #Onlu sayı
print(mystr6.isnumeric()) #Sayısal değer
```

```
False
True
False
False
```

```
#Escape (Kaçış) Karakteri
mystr10 = "My favourite TV Series is "Game of Thrones""
print(mystr10)
```

```
  File "<ipython-input-197-d812e4f3385e>", line 2
    mystr10 = "My favourite TV Series is "Game of Thrones""
```

```
                                                  ^
SyntaxError: invalid syntax
```

```
mystr10 = "My favourite TV Series is \"Game of Thrones\""
print(mystr10)
```

```
My favourite TV Series is "Game of Thrones"
```

```
hello = 'hello'
python = 'python'

print(hello + " " + python)
print("%s %s" % (hello, python))
print("{} {}".format(hello, python))
print(' '.join([hello, python]))
```

```
hello python
hello python
hello python
hello python
```

```
#Sözlükler (Dictionaries)

mydict = dict()
mydict
```

```
{}
```

```
mydict = {}
mydict
```

```
{}
```

```
mydict = {1 : 'one', 2: 'two', 3: 'three'}
mydict
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
mydict = dict({1 : 'one', 2: 'two', 3: 'three'})
mydict
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
mydict = {'A' : 'one', 'B': 'two', 'C': 'three'}
mydict
```

```
{'A': 'one', 'B': 'two', 'C': 'three'}
```

```
mydict.keys()
```

```
dict_keys(['A', 'B', 'C'])
```

```
mydict.values()
```

```
dict_values(['one', 'two', 'three'])
```

```
mydict.items()
```

```
dict_items([('A', 'one'), ('B', 'two'), ('C', 'three')])
```

```
mydict = {1 : 'one', 2 : 'two', 'A' : ['Doğu', 'Batı', 'Kuzey']}
mydict
```

```
{1: 'one', 2: 'two', 'A': ['Doğu', 'Batı', 'Kuzey']}
```

```
mydict = {1 : 'one', 2 : 'two', 'A' : ['Doğu', 'Batı', 'Kuzey'], 'B' :
['bat', 'hat', 'cat']}
mydict
```

```
{1: 'one',
 2: 'two',
 'A': ['Doğu', 'Batı', 'Kuzey'],
 'B': ['bat', 'hat', 'cat']}
```

```
keys = {'a', 'b', 'c', 'd'}
mydict3 = dict.fromkeys(keys)
mydict3
```

```
{'b': None, 'd': None, 'c': None, 'a': None}
```

```
keys = {'a', 'b', 'c', 'd'}
value = 10
mydict3 = dict.fromkeys(keys, value)
mydict3
```

```
{'b': 10, 'd': 10, 'c': 10, 'a': 10}
```

```
keys = {'a', 'b', 'c', 'd'}
value = [10, 20, 30]
mydict3 = dict.fromkeys(keys, value)
mydict3
```

```
{'b': [10, 20, 30], 'd': [10, 20, 30], 'c': [10, 20, 30], 'a': [10,
20, 30]}
```

```
value.append(40)
mydict3
```

```
{'b': [10, 20, 30, 40],
 'd': [10, 20, 30, 40],
 'c': [10, 20, 30, 40],
 'a': [10, 20, 30, 40]}
```

```
#Elemanlara Erişim
```

```
mydict = {1 : 'one', 2: 'two', 3: 'three', 4: 'four'}
mydict
```

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
mydict[1]
```

```
'one'
```

```
mydict.get(1)
```

```
'one'
```

```
mydict1 = {'Name' : 'Doğu', 'ID' : 74123, 'DOB' : 1980, 'job' :
'Analyst'}
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'job': 'Analyst'}
```

```
mydict1['Name']
```

```
'Doğu'
```

```
mydict1.get('job')
```

```
'Analyst'
```

```
#Add, Remove, Change
```

```
mydict1 = {'Name' : 'Doğu', 'ID' : 74123, 'DOB' : 1980, 'Address' :
'Stockholm'}
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'Address': 'Stockholm'}
```

```
mydict1['DOB'] = 1982
mydict1['Address'] = 'Roma'
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1982, 'Address': 'Roma'}
```

```
mydict1['job'] = 'Analyst'
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1982, 'Address': 'Roma', 'job':
'Analyst'}
```

```
mydict1.pop('job')
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1982, 'Address': 'Roma'}
```

```
mydict1.popitem()
```

```
('Address', 'Roma')
```

```
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1982}
```

```
del mydict1['ID']
mydict1
```

```
{'Name': 'Doğu', 'DOB': 1982}
```

```
mydict1.clear()
mydict1
```

```
{}
```

```
del mydict1
mydict1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-232-a5f568839d89> in <module>
      1 del mydict1
----> 2 mydict1

NameError: name 'mydict1' is not defined
```

```
#Kopyalama
```

```
mydict5 = {'Name' : 'Doğu', 'ID' : 74123, 'DOB' : 1980, 'Address' : 'Stockholm'}
mydict5
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'Address': 'Stockholm'}
```

```
mydict1 = mydict5
```

```
id(mydict1), id(mydict5)
```

```
(2362671418816, 2362671418816)
```

```
mydict2 = mydict5.copy()
```

```
id(mydict2)
```

```
2362670484160
```

```
mydict5['Address'] = 'Ankara'
```

```
mydict5
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'Address': 'Ankara'}
```

```
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'Address': 'Ankara'}
```

```
mydict2
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'Address': 'Stockholm'}
```

```
#Döngüyle İterasyon

mydict1 = {'Name' : 'Doğu', 'ID' : 74123, 'DOB' : 1980, 'Address' :
'Stockholm', 'Job' : 'Analyst'}
mydict1
```

```
{'Name': 'Doğu',
 'ID': 74123,
 'DOB': 1980,
 'Address': 'Stockholm',
 'Job': 'Analyst'}
```

```
for i in mydict1:
    print(i, ':', mydict1[i])
```

```
Name : Doğu
ID : 74123
DOB : 1980
Address : Stockholm
Job : Analyst
```

```
for i in mydict1:
    print(mydict1[i])
```

```
Doğu
74123
1980
Stockholm
Analyst
```

```
#Membership (Üyelik)

mydict1 = {'Name' : 'Doğu', 'ID' : 74123, 'DOB' : 1980, 'job' :
'Analyst'}
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'job': 'Analyst'}
```

```
'Name' in mydict1
```

```
True
```

```
'Doğu' in mydict1
```

```
False
```

```
'ID' in mydict1
```

```
True
```

```
'Address' in mydict1
```

```
False
```

```
#All, Any
```

```
mydict1 = {'Name' : 'Doğu', 'ID' : 74123, 'DOB' : 1980, 'job' :
'Analyst'}
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'job': 'Analyst'}
```

```
all(mydict1)
```

```
True
```

```
any(mydict1)
```

```
True
```

```
mydict1[0] = 'test1'
mydict1
```

```
{'Name': 'Doğu', 'ID': 74123, 'DOB': 1980, 'job': 'Analyst', 0:
'test1'}
```

```
all(mydict1)
```

```
False
```

```
any(mydict1)
```

```
True
```

```
#Dictionary Comprehension
```

```
double = {i : i*2 for i in range(10)}
double
```

```
{0: 0, 1: 2, 2: 4, 3: 6, 4: 8, 5: 10, 6: 12, 7: 14, 8: 16, 9: 18}
```

```
square = {i : i**2 for i in range(10)}
square
```

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

```
key = ['one', 'two', 'three', 'four', 'five']
value = [1,2,3,4,5]
```

```
mydict = {k : v for (k, v) in zip(key, value)}
mydict
```

```
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
```

```
mydict1 = {'a' : 10, 'b' : 20, 'c' : 30, 'd' : 40, 'e' : 50}
mydict1 = {k : v / 10 for (k, v) in mydict1.items()}
mydict1
```

```
{'a': 1.0, 'b': 2.0, 'c': 3.0, 'd': 4.0, 'e': 5.0}
```

```
str1 = "Natural Language Processing"
mydict2 = {k : v for (k, v) in enumerate(str1)}
mydict2
```

```
{0: 'N',
 1: 'a',
 2: 't',
 3: 'u',
 4: 'r',
 5: 'a',
 6: 'l',
 7: ' ',
 8: 'L',
 9: 'a',
 10: 'n',
 11: 'g',
 12: 'u',
 13: 'a',
 14: 'g',
 15: 'e',
 16: ' ',
 17: 'P',
 18: 'r',
 19: 'o',
 20: 'c',
 21: 'e',
 22: 's',
 23: 's',
 24: 'i',
 25: 'n',
 26: 'g'}
```

```
str1 = "abcdefghijklmnoprstuvwxyz"
mydict3 = {i : i.upper() for i in str1}
mydict3
```

```
{'a': 'A',
 'b': 'B',
 'c': 'C',
 'd': 'D',
```

```
 'e': 'E',
 'f': 'F',
 'g': 'G',
 'h': 'H',
 'i': 'I',
 'j': 'J',
 'k': 'K',
 'l': 'L',
 'm': 'M',
 'n': 'N',
 'o': 'O',
 'p': 'P',
 'r': 'R',
 's': 'S',
 't': 'T',
 'u': 'U',
 'v': 'V',
 'w': 'W',
 'x': 'X',
 'y': 'Y',
 'z': 'Z'}
```

```
#Sözlük kullanarak kelime sıklığı bulma

mystr4 = "one two three four one two two three five five six seven six
seven one nine nine eight"

mylist = mystr4.split()
mylist
```

```
['one',
 'two',
 'three',
 'four',
 'one',
 'two',
 'two',
 'three',
 'five',
 'five',
 'six',
 'seven',
 'six',
 'seven',
 'one',
 'nine',
 'nine',
 'eight']
```

```
mylist1 = set(mylist)
mylist1 = list(mylist1)
mylist1
```

```
['two', 'five', 'one', 'four', 'six', 'nine', 'eight', 'seven',
'three']
```

```
count1 = [0] * len(mylist1)
mydict5 = dict()
for i in range(len(mylist1)):
    for j in range(len(mylist)):
        if mylist1[i] == mylist[j]:
            count1[i] += 1
    mydict5[mylist1[i]] = count1[i]
print(mydict5)
```

```
{'two': 3, 'five': 2, 'one': 3, 'four': 1, 'six': 2, 'nine': 2,
'eight': 1, 'seven': 2, 'three': 2}
```

```
#Kümeler (Sets)
```

```
myset = {1,2,3,4,5}
myset
```

```
{1, 2, 3, 4, 5}
```

```
len(myset)
```

```
5
```

```
my_set = {1,1,2,2,3,4,5,5}
my_set
```

```
{1, 2, 3, 4, 5}
```

```
myset1 = {1.79, 2.08, 3.99, 4.56, 5.45}
myset1
```

```
{1.79, 2.08, 3.99, 4.56, 5.45}
```

```
myset2 = {'Doğu', 'Batı', 'Kuzey'}
myset2
```

```
{'Batı', 'Doğu', 'Kuzey'}
```

```
myset3 = {10, 20, "Hola", (11,22,33)}
myset3
```

```
{(11, 22, 33), 10, 20, 'Hola'}
```

```
myset3 = {10, 20, "Hola", [11, 22, 33]} #kümeler mutable ögelere izin
vermez!
myset3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-281-663e26dcda0d> in <module>
----> 1 myset3 = {10, 20, "Hola", [11, 22, 33]}
      2 myset3

TypeError: unhashable type: 'list'
```

```
myset4 = set()
print(type(myset4))
```

```
<class 'set'>
```

```
my_set1 = set(('one', 'two', 'three', 'four'))
my_set1
```

```
{'four', 'one', 'three', 'two'}
```

```
#İterasyon
myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight'}

for i in myset:
    print(i)
```

```
two
five
one
four
six
eight
seven
three
```

```
for i in enumerate(myset):
    print(i)
```

```
(0, 'two')
(1, 'five')
(2, 'one')
(3, 'four')
(4, 'six')
(5, 'eight')
(6, 'seven')
(7, 'three')
```

```
myset
```

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
'one' in myset
```

```
True
```

```
'ten' in myset
```

```
False
```

```
if 'eleven' in myset:
    print("Eleven is present in the set")
else:
    print("Eleven is not present in the set")
```

```
Eleven is not present in the set
```

```
#Add, Remove
```

```
myset
```

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
myset.add('NINE')
myset
```

```
{'NINE', 'eight', 'five', 'four', 'one', 'seven', 'six', 'three',
'two'}
```

```
myset.update(['TEN', 'ELEVEN', 'TWELVE'])
myset
```

```
{'ELEVEN',
 'NINE',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}
```

```
myset.remove('NINE')
myset
```

```
{'ELEVEN',
 'TEN',
 'TWELVE',
 'eight',
 'five',
 'four',
```

```
 'one',
 'seven',
 'six',
 'three',
 'two'}

myset.discard('TEN')
myset

{'ELEVEN',
 'TWELVE',
 'eight',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'three',
 'two'}

myset.clear()
myset

set()

del myset
myset

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-297-0912ea1b8932> in <module>
      1 del myset
----> 2 myset

NameError: name 'myset' is not defined

#Kopyalama
myset = {'one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight'}
myset

{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

myset1 = myset
myset1

{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

id(myset), id(myset1)
```

```
(2362671404640, 2362671404640)
```

```
my_set = myset.copy()
my_set
```

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
id(my_set)
```

```
2362671404192
```

```
myset.add('nine')
myset
```

```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three',
'two'}
```

```
myset1
```

```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three',
'two'}
```

```
my_set
```

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
#Küme Operasyonları
```

```
#Birleşim
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

```
A | B
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
A.union(B)
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
A.union(B, C)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
A.update(B, C)
```

```
A
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
#Kesişim
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

```
A & B
```

```
{4, 5}
```

```
A.intersection(B)
```

```
{4, 5}
```

```
A.intersection_update(B)
A
```

```
{4, 5}
```

```
#Fark (Difference)
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

```
A - B
```

```
{1, 2, 3}
```

```
A.difference(B)
```

```
{1, 2, 3}
```

```
B - A
```

```
{6, 7, 8}
```

```
B.difference(A)
```

```
{6, 7, 8}
```

```
B.difference_update(A)
```

```
B
```

```
{6, 7, 8}
```

```
#Simetrik Fark (A - B) U (B - A)
```

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
```

```
A ^ B
```

```
{1, 2, 3, 6, 7, 8}
```

```
A.symmetric_difference(B)
```

```
{1, 2, 3, 6, 7, 8}
```

```
A.symmetric_difference_update(B)
A
```

```
{1, 2, 3, 6, 7, 8}
```

```
#Superset, Subset, Disjoint

A = {1,2,3,4,5,6,7,8,9}
B = {3,4,5,6,7,8}
C = {10,20,30,40}

B.issubset(A)
```

```
True
```

```
A.issuperset(B)
```

```
True
```

```
C.isdisjoint(A) #birbirleriyle hiç ortak elemanlarının olmaması durumu
```

```
True
```

```
B.isdisjoint(A)
```

```
False
```

```
#Diğer Dahili (Built-in) Fonksiyonlar

A
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
sum(A)
```

```
45
```

```
max(A)
```

```
9
```

```
min(A)
```

```
1
```

```
len(A)
```

```
9
```

```
list(enumerate(A))
```

```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
9)]
```

```
D = sorted(A, reverse = True) #descending
D
```

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

```
sorted(D) #ascending
```

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

```
#Fonksiyonlar

"""
def myfunc(positional arguments, *args (tuple), named arguments,
**kwargs (dictionary))
def myfunc(120, 160, 50, b = 10, c = 20, a = 30)
"""
```