

👉 Bu dosya **Mahmut Edip Negiz** tarafından oluşturulmuştur.

👉 iletişim için *mahmutedipnegiz@gmail.com*

👉 LinkedIn <https://www.linkedin.com/in/mahmutedipnegiz/>

👉 Github <https://github.com/medipnegiz>

STRING İFADE KOD KULLANIMLARI

```
In [ ]: konu = '  Havalar iyice, sogudu Murat beyyy  '
text = 'Muratterzi'
text += ' adam'
print('1:', konu.swapcase())
print('2:', konu.upper())
print('3:', konu.lower())
print('4:', konu.casefold())
print('5:', konu.capitalize())
print('6:', konu.title())
print('7:', konu.replace('r', 'g', 2))
print('8:', konu.count('a', 5, 25))
print('9:', konu.startswith('Hava', 2))
print('10:', konu.endswith('bey', 0, 32))
print('11:', konu.startswith('i', 8))
print('12:', konu.strip(' y '))
print('13:', konu.lstrip(' Ha'))
print('14:', konu.rstrip('ey '))
print('15:', konu.find('a'))
print('16:', konu.rfind('t'))
print('17:', text.index('a'))
print('18:', *konu)
print('19:', 'iyi' in konu)
print('20:', '@' not in konu)
print('21:', '#'.join(konu))
print('22:', konu.center(46, '+'))
print('23:', konu.ljust(40, '.'))
print('24:', konu.rjust(50, '/'))
print('25:', konu.zfill(45))
print('26:', konu.split('a'))
print('27:', konu.split(',')[0])
print('28:', min(konu), max(konu))
print('29:', text.isalpha())
print('30:', text.isdigit())
print('31:', konu.isalnum())
print('32:', text)
```

string ifade olara
büyük ve küçük har
hepsini büyük yapa
hepsini küçük yapa
hepsini küçük yapa
Sadece satir basi
Tüm kelimeler büyü
cümle icinden iste
cümlede istenilen
cümle istenilen ke
cümle istenilen ke
cümle icinden konu
cümlelerin iki taraf
soldan istedigimiz
sagdan istedigimiz
cümle icinden iste
cümle icinden iste
cümle icinden iste
" * " harflerin ar
icinde istenilen h
icinde istenilen h
her karakter sonra
istenilen karakter
yaziyi sola yaslar
yaziyi saga yaslar
yazinin basina ist
ayristirma yapiyor
icine girilen kara
ASCII kod tablosun
tamamı harflerden
tamamı rakamlardan
rakam ve harften m

SATIR ve METİN DÜZENİ

```
In [ ]: print('1:', ' Oyun ', ' sahasi ', ' kapali ', sep='+')
print('2:', ' Oyun ', ' sahasi ', ' kapali ', end='$')
print('3:', '\t', ' Oyun ', ' sahasi ', ' kapali')
print('4:', '\n', 'Oyun ', ' sahasi ', ' kapali')
print('5:', ' Oyun ', ' sahasi ', ' kapali', '\b\b')
print('6:', 'Merhaba Arkadas nasilsin \r Zalim Dünya!')
```


#

TRUE ve FALSE İFADELER

```
In [ ]: print('1:', False or True)           # " or " toplama islemi gibi
        print('2:', True and False)        # " and " carpma islemi gibi
        print('3:', None or '')           # ici bos tirnaklar 0 kabul
        print('4:', 'a' and not True)      # " not " ifadesi tam tersin
        print('5:', 0 or 1)                # 0 ve 1 False ve True olara
        print('6:', True * True)           # carpm islemi gibi yazdira
        a, b = 5, 7
        print('7:', id(a), id(b))          # tanimli ifadenin bellek ad
        print('8:', a is b)                # Bellek adresleri ayni mi d
        print('9:', 'ali' or False)        # Tirnak icine yazilan her i
                                           # Islem sirasi: parantez > n
                                           # ' ', '"', '{ } bu degerler Fal
```

MATEMATİKSEL İSLEMLER

```
In [ ]: from math import sqrt

        print('1:', 36 // 5)                # | "+=" --> "x = x + n" | ## | "-
        print('2:', 36 / 5)                 # bölme islemi yapar tam sayi ve
        print('3:', 36 + 5)                 # bölme islemi yapar küsuratli v
        print('4:', 36 - 5)                 # toplama islemi yapar
        print('5:', 36 % 5)                 # cikarma islemi yapar
        print('6:', 2 ** 5)                 # bölme isleminden kalani verir
        print('7:', 9 ** 0.5)               # üssünü alma islemidir
        print('8:', int(5.0))               # Karekök alma yöntemidir. Kesir
        print('9:', str(5.0))               # " int " tam sayi yapar
        print('10:', float(5))              # " str " metin yapar
        print('11:', complex(3.45j))        # " float " kesirli sayi yapar
        print('12:', bool(1))               # " complex " karisik sayi yapar
        print('13:', 5 == 5)                # " bool " icindeki ifadenin Tru
        print('14:', 6 != 5)                # denktir isareti
        print('15:', 6 is 5)                # denk degildir isareti
        print('16:', round(16.65754, 2))    # esittir anlami vardir
        print('17:', round(123_675, -3))    # sayiyi en yakin degere yuvarla
        print('18:', bin(12))               # virgülden sonra negatif sayi g
        print('19:', oct(12))               # sayinin ikili sistemdeki karsi
        print('20:', hex(12))               # sayinin sekizli sistemdeki kar
        print('21:', pow(3, 2))             # sayinin onaltili sistemdeki ka
        print('22:', sqrt(9))               # sayinin üssünü almak icin kull
        a = 5
        print('23:', eval('pow(a,2)'))      # sayinin karekökünü alır. Math .
                                           # icine girilen "a" degeri tanim
```

DILIMLEYEREK YAZDIRMA KULLANIMI

```
In [ ]: city = 'Dresden'

        print('1:', city[1:])               # belirledigimiz yerden itibaren y
        print('2:', city[:4])               # belirlenen yere kadar yazar (son
        print('3:', city[:2])               # tamamini 2 satir atlayarak yazar
        print('4:', city[-5:])              # normal yazar sadece eksili ifade
        print('5:', city[:-1])              # tamamini tersten yazar
        print('6:', len(city))              # icinde kac tane karakter var onu
        print('7:', type(city))             # tanimli ögenin tipini verir
```

```
In [ ]: cumle = 'Dun %d kilo %s aldim ama %.2f grami bozuktuk ' % (5, 'tavuk', 50)
print('1:', cumle)

print('2:', '%.15s' % cumle)

sayi = 12345
print('3:', '%.7d' % sayi)

print('4:', '%(kilo)d kilo %(meyve)s aldim' % {'kilo': 3, 'meyve': 'portakal'})

print('5:', format("Welcome", "10s"), end = '#')
print('6:', format(111, "4d"), end = '#')
print('7:', format(924.656, ".2f"))
```

FORMAT YÖNTEMİ KULLANIMI

```
In [ ]: meyve = 'Karpuz'
miktar = 5
print('1:', 'Ali {} kiloluk {} aldi'.format(miktar, meyve))

print('2:', 'hava {0} ama {1} hep {isi} dereceydi'.format(36, 'evimiz', is))

print('3:', '{2} {3} {1} {0} {4}'.format('ali', 22, 'yasina', 'geldigi', 'yeni'))

yeni = f'satilan {meyve}un {miktar} kilosu cöpe gitti'
print('4:', yeni)

yazi = '%d kilo %s satin aldim' % (3, 'elma')
print('5:', yazi)

num = 3.1456789
print('6:', 'Pi sayisi: {:07.3f}'.format(num))

num2 = 9876543214
print('7:', f'{num2:,}')
```

LIST FORMATI KOD KULLANIMLARI

```
In [ ]: matrix = [
    [1,2,3,4],
    [5,6,7,8],
    [9,1,2,3],
]

nummer = [2, 4, 9, 13, 1, 3, 21, 3] # Kösel
num = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
ulke = ('Paris', 'Italya', 'Prag')
yeni_list = [ulke] # Liste
dil = [['Deutsch'], ['Englisch', 'Türkisch'], ['Italianisch']]
sehir = ['Berlin', 'Köln', 'Dresden', 'Frankfurt'] # Kösel
plaka = [35, 'KL', 'D', 15]
nums = list(range(1,30,2)) # Veril
satir = list(zip(*matrix)) # " zip
birles = matrix[0] + matrix[1] + matrix[2] # liste
a, b, c = matrix # Liste
new_list = sorted(dil) # Yeni
ulke_yeni = list(ulke) # bu se
sehir.append('Ankara') # Liste
plaka.extend(sehir) # Paran
sehir.insert(1, 'Izmir') # Liste
```

```

nummer.pop(1) # Veril
sehir.remove('Köln') # Liste
sehir.sort() # ASCII
sehir.reverse() # Terst
#ulke_yeni.clear() # Liste
print('1:', len(ulke)) # Liste
print('2:', dil[0][1][1]) # Her k
ulke_yeni[1] = 'Amerika' # Liste
print('3:', sehir, '\n' + '4:', ulke_yeni)
print('5:', type(ulke), '\n' + '6:', yeni_list, len(yeni_list))
print('7:', ''.join(ulke)) # " joi
print('8:', dir(list)) # " dir
print('9:', new_list, '\n' + '10:', satir)
print('11:', birles, '\n' + '12:', a+b+c)
print('13:', *plaka) # " * "
print('14:', nummer, sum(nummer)) # " Sum
print('15:', nums)
print('16:', min(nummer)) # Liste
print('17:', max(nummer)) # Liste
print('18:', max(num, key = num.count)) # "key"
print('19:', min(ulke, key = len)) # karak
print('20:', list(enumerate(sehir,1))) # Veril

name, num, bos = ['ali', 'veli'], ['can', False], []
print('21:', all(name), all(num), all(bos)) # Yinel
print('22:', any(name), any(num), any(bos)) # Yinel

number = iter([256, 'Ahmet', 12]) # Yinel
print('23:', next(number, 'dur')) # "next
print('23:', next(number, 'dur')) # Nesne
print('23:', next(number, 'dur')) # Bu ko
print('23:', next(number, 'dur')) # Sonda

```

TUPLE (DEMET) FORMATI KULLANIMLARI

```

In [ ]: # *** icerigi degistirilemez, Listelerden daha hizli calisirlar ve az ye
t = () # parantez icine
t1 = tuple() # "tuple" sarmal
t2 = 'p', 'y', 't', 'h', 'o', 'n' # parantez olmas
t3 = 1, 2, 3, 4 # sayilarda virg
t4 = ('python') # parantez icind
t5 = ('a', 'b') # virgül ile ayr
t6 = ['Mehmet']
t7 = tuple(t6) # listeler de "t
t8 = t2, t3, t5 # ic ice tuple o
a, b, c = t8 # burada tuple i
print('1:', 't:', type(t), '\nt1:', type(t1))
print('2:', 't2:', type(t2), '\nt3:', type(t3))
print('3:', 't4:', type(t4), '\nt5:', type(t5))
print('4:', t7) # "tuple" ciktil
print('5:', *t5) # " * " isareti
print('6:', t8)
print('7:', t8[0]) # ic ice tuple f
print('8:', [1, 2, 3], [3, 4], [5][0]) # ic ice girerek
print('9:', b)
print('9:', tuple(enumerate(t2))) # tuple formatin

```

DICT(SÖZLÜK) FORMATI KOD KULLANIMLARI

In []:

```

form = {'personal':{
    'erkek':
        {'ali': {'sname': 'Kurt', 'age': 25},
         'veli': {'sname': 'Nergiz', 'age': 32},
        },
    'bayan':
        {'ayse': {'sname': 'Ulu', 'age': 21},
         'umran': {'sname': 'Demir', 'age': 19}
        },
    },
}

ad = [ 'one', 'two', 'three', 'four', 'five']
no = [1, 2, 3, 4, 5]
mix_szlk = {'animal': 'dog', 'planet': 'neptun', 'number': 40, 'good': True}
my_dict = {1: 'bir', 2: 'iki', 3: 'üç', 4: 'dört', 5: 'bes', 6: 'altı'}
szlk_atama = dict(one = 1, two = 2, three = 3)
mix_szlk['land'] = 'Italia'
mix_szlk['planet'] = 'Mars'
mix_szlk.update({'good': False})
#mix_szlk.clear()
grocer = dict(meyve = 'Armut', sebze = 'Domates')
sozluk = dict(zip(ad, no))
sozluk_list = list(sozluk.items())
print('1:', form['personal']['erkek']['ali']['sname'])
print('2:', form['personal'].items())
print('3:', *form['personal']['erkek'].keys())
print('4:', form['personal']['bayan']['umran'].values())
del form['personal']['erkek']
print('5: silindi..:', mix_szlk.pop('number', 'Sözlükte yok'))
#print('6: sildik..:', mix_szlk.popitem())
print('7:', form['personal'])
print('8:', sozluk, '\n9:', grocer)
print('10:', sozluk_list, '\n11:', mix_szlk)
print('12:', szlk_atama)
print('13:', 'animal' in mix_szlk)
print('14:', 'dog' not in mix_szlk.values())
rakam = 2
print('15:', my_dict.setdefault(7, 'icinde yok'), my_dict)
print('16:', my_dict.get(rakam, 'rakam bulunamadi'))

```

SET (KÜME) FORMATI KOD KULLANIMLARI

```

In [ ]: a = {'orange', 'blau', 'red', 'blau', 'rose'} # "{}"
flower = 'rose', 'orkhide', 'rose', 'violet', 'red' # "set"
b = set(flower) # bos
bos_kume = set() # bos
c = set('philadelphia') # icin
d = set('dlia') # icin
e = {'dolphin'} # Iki
print('1:', a.difference(b)) # Iki
print('2:', b.difference(a)) # Iki
print('3:', a - b) # Iki

print('4:', a.union(b)) # Iki
print('5:', a | b ) # Iki

print('6:', a.intersection(b)) # Sade
print('7:', a & b) # Sade
bos_kume = frozenset(bos_kume) # "fro
a.remove('orange') # Küme
new = a.copy() # A kü

```

```

a.add('yellow') # Küme
print('8:', a)
print('9:', 'red' in b) # Küme
print('10:', 'purple' not in b) # Küme
print('11:', c)
print('12:', e)
print('13: silindi...', c.pop()) # küme
c.discard('p') # küme
print('14:', d.issubset(c)) # d kü
print('15:', c.issuperset(d)) # c kü

```

IF ve INPUT KOMUT KULLANIMLARI

```

In [ ]: score = int(input('Puaninizi girin..: ')) # kullanıcı tarafından g
if score >= 90: # her zaman bir sarta ba
    if score >= 95: # ic ice if komutu veril
        derece = 'A+' # if tanim ve yazdirmada
    else:
        derece = 'A'
elif score >= 80: # "elif" bu komutun "ege
    derece = 'B'
elif score > 70 or score == 70: # "or" komutu ile ikisin
    derece = 'C'
elif score >= 50 and score <= 60: # "and" komutu ile iki d
    pass # "pass" komutu herhangi
elif score == 40:
    raise Exception('Bu nasıl not...:') # "raise" istenmeyen bir
else: # bu komutun "bunların h
    derece = 'Gecemediniz'
print('Dereceniz..: ', derece)

sayi = []
for i in range(10):
    if i % 2 == 0: # Çift sayıları listeye
        sayi.append(i)
    else:
        sayi.append(i**2)
print('1:', sayi)

yeni = [i if i % 2 == 0 else i**2 for i in range(10)] # for, i
print('2:', yeni)

sayi = 50
print('3:', 'Sayı 50' if sayi == 50 else 'Sayı belirsiz') # Tek sa

```

FOR DÖNGÜ KOMUT KULLANIMI

```

In [ ]: # For döngüsü genellikle liste, tanımlama grubu, küme ve sözlük gibi çeşitli
season = ['spring', 'summer', 'autumn', 'winter']
for a in season: # liste içinde gezer
    print('1:', a)

dongu = list(range(1,8)) # " range " istenile
para = 1000
for i in dongu: # liste içinde döngü
    para += para * (0.07)
    i += 1 # burada "i" tanımla
print('2:', int(para), '$') # print boşlukları ö

```

```

print('3:', dongu)

who = ['Ich bin ', 'Du bist ']
mood = ['froh', 'angry', 'müde']
for i in who:
    for c in mood:
        print('4:', i, c)

text = ['eins', 'zwei', 'drei']
num = [1, 2, 3, 4]
for x, y in zip(text, num):
    print('5:', x, ':', y)

print('6:', *range(16,0,-2))

for i in "pythontr":
    if i == "t":
        continue
    print(i, end=' ')

cift = []
for i in range(12):
    if i % 2 == 0:
        cift.append(i*i)
print('\n8:', cift)

yeni = [i*i for i in range(12) if i % 2 == 0]
print('9:', yeni)

```

ic ice for döngüsü
dis döngü baslar i

"zip" komutu ile s
döngü icerisinden

verilen degere gör

burada eger kosul

cift sayilarin kar

for ve if komutlar
asil islem basa ya

WHILE DÖNGÜ KOMUT KULLANIMI

In []:

```

liste = ['a', 'b', 'c', 'd', 'e']
a = 0
while a < len(liste):
    print('1: {} karesi...: {}'.format(a, a**2))
    a += 1

sayi = 45
while True:
    tahmin = int(input('sayi kac...: '))
    if tahmin < sayi:
        print('2: daha büyük')
    elif tahmin > sayi:
        print('3: daha küçük')
    else:
        print('4: sayiyi buldun')
        break

yas = input('Yasinizi girin...:')
while not yas.isdigit():
    yas = input('Lütfen rakam olarak girin...: ')
print('5: Bravo yasin :', yas)

```

bu komut kosula ba

her seferinde list

True ifadesi ile s

döngü icerisinde "

döngüden cikmak ic

"isdigit" komutu s

DEF FONKSİYONU KULLANIMI

In []:

```

b = 5
def deger():
    global b
    b = 7

```

burada "b" global bir degisk

"def" komutu ile fonksiyon b

fonksiyondaki "b" degiskenin

fonksiyon icinde deger atama

```

    #print(b)
deger()
print('1:', b)

def sayi():
    s = 5
    return s
son = sayi()
print('2:', sayi())
print('3:', son)

def coklu(sayi, mesaj = 'Hallo', dongu = 3):
    print('4:', sayi, 'kez', mesaj, dongu, 'e kadar')
coklu(5)
coklu(4, 'Naberr')
coklu(mesaj='Gece', sayi=8, dongu=10)

def meyveler(*fruit):
    print('5:', fruit)
hazir = ['Cilek', 'Karpuz', 'Erik']
meyveler('Portakal', 'Elma', 'Armut')
meyveler(*hazir)

def hayvan(**animal):
    for i, k in animal.items():
        print('6:', k, 'bir', i)
aile = {'y': 'Murat', 'x': 'Ufuk'}
hayvan(etobur='Aslan', otobur='Fil', hepobur='Ayi')
hayvan(**aile)

number = [1,2,5,8,9,11,12,16,20,25,37,38]
def cift(x):
    return x % 2 == 0
print('7:', list(filter(cift, number)))

def kare(sayi):
    return sayi * sayi
sayi = range(1,10)
print('8:', list(map(kare, sayi)))

def ana():
    x = 'Dis katman'
    def icten():
        nonlocal x
        x = 'Ic katman'
        print('9: Icte: ', x)
    icten()
    print('9: Dista: ', x)
ana()

def position(konum, /):
    print(konum)
def position(*, keyword=5):
    print(keyword)

def faktoriyel(x):
    if x <= 1:
        return 1
    else:
        return x * faktoriyel(x-1)
print('10: ', faktoriyel(5))

import keyword
print('11', keyword.kwlist, end=' ')

```



```
print('12: ', bool.__doc__)
help(pow)
```

```
#  "__doc__"
#  "help" önc
```

LAMBDA FONKSİYONU KULLANIMI

In []:

```
print('1:', (lambda x: x**2)(3)) # "x
print('2:', (lambda x, y: (x + y) / 2) (9, 7)) # bi

tek_cift = (lambda x: 'cift' if x % 2 == 0 else 'tek') # bi
print('3:', tek_cift(16)) # pr
print('4:', (lambda x: 'cift' if x % 2 == 0 else 'tek')(16)) # di

sayi = [2, 3, 4, 5, 6, 7, 8, 9]
sonuc = list(map(lambda x: x**2, sayi)) # "m
print('5:', sonuc) # **
print('6:', list(map(lambda x: x**2, sayi))) # ya

print('7:', list(filter(lambda i: i % 2 == 0, sayi))) # "f
# "f

liste = [None, '', 0, False, 'Fresh', 'Ahmet', None, 'Veli']
print('8:', list(filter(bool, liste))) # Eg
print('9:', list(filter(None, liste))) # Eg

sesli = ['a', 'e', 'i', 'o', 'u']
harfler = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
s = filter(lambda k: True if k in sesli else False, harfler) # "f
print('10: Sesli harfler:', list(s)) # li

def carp(n):
    return lambda a: a * n # de
kere = carp(3) # de
print('11:', kere('Ufuk ')) # la

from functools import reduce # "r
number = [11, 23, 9, 34, 2, 64] # "r
sonuc = lambda x, y: x if x > y else y # la
print('12: En büyük sayı:', reduce(sonuc, number)) # **

print('13:', reduce(lambda a, b: a + b, range(101))) # "r
print('14:', reduce(lambda x,y: x+y, [[1,2,3],[4,5],[6,7,8]])) # "r
```

MODÜLLER

In []:

```
### Önceden oluşturduğumuz bir ".py" uzantili python dosyamızı, içindeki f
### Proje ve önceden oluşturulan ".py" uzantili dosya aynı klasör içerisinde

import my_module # modülümüzü projeye
my_module.fonksiyonum() # modülün içerisinde
print(my_module.sayac) # modül içerisindeki

import my_module as sözlük # import edilen bir
sözlük.fonksiyonum() # takma isim ile mod
print(sözlük.sayac) # takma isim ile mod

from my_module import fonksiyonum # modül içerisinden
fonksiyonum() # sadece fonksiyon c

from my_module import fonksiyonum as kare # modül içerisinden
kare() # takma isim ile fon
```

```

from math import pi, factorial, log10                                # aynı anda birden fazla import
import string
print(string.punctuation)                                           # mevcut tüm noktalar
print(string.digits)                                                # tüm rakamları yazdır
print(string.ascii_letters)                                         # ascii tablosunda olan harfleri
print(string.ascii_lowercase)                                       # ascii tablosunda olan küçük harfleri
print(string.ascii_uppercase)                                       # ascii tablosunda olan büyük harfleri
print(string.printable)                                             # yazdırılabilir tüm karakterleri

import datetime
print(datetime.date.today())                                       # bugünün tarihini yazdır
print(datetime.datetime.now())                                     # bugünün tarihini ve saatini yazdır

import calendar
calendar.prcal(2022)                                                # belirli bir yılını gösterir
calendar.prmonth(2022, 4)                                           # sadece belirli bir ayı gösterir

my_module.__doc__                                                  # dosyanın içerisindeki dokümantasyon
__all__ = ['kare_alme', 'us_alma']                                  # dosyanın dışına çıkarılacak değişkenler

if __name__ == '__main__':                                         # bu kısım programa başlatıldığında çalışır
    print('Burasi modülde görünmez')                                # programı kendi başlatırken çalışmaz
else:                                                                # "else" komutu eklenirse çalışır
    print('Burasi modülde görünür')                                # ana program çalışırken çalışır

```

PAKETLER

```

In [ ]: # earth klasörü/                                           # iç içe klasörler olsun
#         __init__.py                                           # her klasörün içerisinde __init__.py dosyası olmalı
#         asia klasörü/
#             __init__.py
#             mongolia.py
#             pakistan.py
#         europe klasörü/
#             __init__.py
#             germany.py
#             kosovo.py
#         america klasörü/
#             __init__.py
#             canada.py
#             ustates.py

# __init__.py dosyası içine --> __all__ = ['europe', 'asia'] gibi
# her alt paketin içine yukarıdaki gibi __init__.py dosyası oluşturularak kullanılabilir

import earth.europe.kosovo                                         # iç içe klasöre girilerek
earth.europe.kosovo.my_function()                                  # kosovo modülündeki fonksiyon çağrılır

from earth import europe.kosovo                                    # alt paket içinden import
europe.kosovo.my_function()                                       # daha kısa yazım ile kullanılabilir

from earth.europe import kosovo                                    # alt paket içerisindeki
kosovo.my_function()                                              # modül içerisindeki fonksiyon çağrılır

from earth.europe.kosovo import my_function                      # sadece fonksiyonu import
my_function()                                                     # fonksiyon import edilmeden kullanılabilir

from earth import *                                                # paketin sahip olduğu tüm modülleri
from earth.europe import *                                         # alt paket içerisindeki tüm modülleri

```

```

from . import mongolia      # mevcut klasörün içinde
from .. import europe      # bir üst klasördeki pak
from ..europe import kosovo # bir üst klasörden bask

dir(europe)                 # paketin içerisindeki m
print(germany.__doc__)      # modülün içerisindeki y

```

PIP KULLANIMI (Pip Installs Packages)

```

In [ ]: pip --version      # su anda yüklü olan pip sürümün
        %pip install my_package # colabda yükleme yapmak için kom
        pip instal my_package  # istenilen bir paketin yüklenme
        pip instal my_package==3.2.1 # paketin sürüm numarası giriler
        pip list               # bilgisayarda kurulu olan tüm p
        pip show my_package    # Paketler hakkında ad, sürüm, ö
        pip uninstall my_package # yüklü olan bir paketi kaldırma

```

RANDOM KOMUTU

```

In [ ]: import random
        a = random.randint(1,20)      # rastge
        b = random.uniform(0.5, 2.0)  # rastge
        c = random.random()           # 0 - 1

        liste = ['ali', 'veli', 'ahmet', 'mehmet', 'celal']
        d = random.choice(liste)       # liste
        e = random.choice(liste[0])    # liste

        f = list(range(11))
        random.shuffle(f)              # Listey

        s = range(100)
        g = random.sample(s, 3)        # belirt

        print('1:', a)
        print('2:', b)
        print('3:', c)
        print('4:', d)
        print('5:', e)
        print('6:', f)
        print('7:', g)

```

EMOJILER ve ASCII KARAKTERLER

```

In [ ]: import time
        print('1:', '\N{Ghost}')      # emojiinin kendi ismi
        print('2:', '\N{shark}')
        print('3:', '\N{spouting whale}')
        print('4:', '\N{front-facing baby chick}')
        print('5:', '\U0001F600')     # emojiinin kodu girile
        print('6:', ord('@'))         # ASCII kodunu öğrenme
        print('7:', chr(64))          # ASCII kod girilir ka
        time.sleep(2)                 # gecikme eklemek için
        print('8:', chr(ord('b') + 1))

```

HATALAR ve HATA MESAJLARI

```
In [ ]: ## SyntaxError --> Yazimda bir hata oldugunda alinan bir

#### EXCEPT HATALARI ####
# TypeError --> Beklenen tür, verilen bir değişken
# ValueError --> Beklenen bir değer verilmediğinde oluş
# NameError --> Tanımsız bir değişkene veya işleve eri
# IOError --> Var olmayan bir dosyaya erişmeye çalış
# IndexError --> Bir dizinin geçersiz bir dizinine eriş
# KeyError --> Sözlükteki bir değere erişmek için geç
# ZeroDivisionError --> Sifira bölmeye calistigimizda alinan b

while True:
    try: # Bu kismin
        sayi1 = int(input('Ilk sayiyi girin...'))
        sayi2 = int(input('Ikinci sayiyi girin...'))
        bol = sayi1 / sayi2
    except Exception: # Her hata t
    except ValueError: # Programda
        print('1: Girilen deger hatali!')
    except (NameError, TypeError): # Parantez i
        print('2: Lütfen rakam girin!')
    except Exception as e: # Hatanin tü
        print(f'3: Muhtemel hataniz...: {e}') # Burada hat
    else: # Program no
        print('4: Herhangi bir hata yok :)')
        break
    finally: # Programda
        print('5: Tesekkürler')

kat = int(input('Kacinci kat...'))
if kat == 13:
    raise NameError('6: Bu kata cikilmaz!!') # Hata türün
print(f'7: Asansör {kat}. kata cikti ')
```

DOSYA OKUMA VE YAZMA ISLEMLERİ

```
In [ ]: # Metin dosyalari --> Metin editörü ile acilan dosyalardir (
# Binary dosyalari --> Metin haricinde "resim, ses, video" ic
# '\n', ' ', None --> Bu degerler satir sonunu belirtir
# Dosya yolu belirtirken --> c:/home/edip/Python/myfile.txt - yada
# pwd --> mevcut dosya konumunu öğrenmek için ku
# ls --> mevcut dizindeki varolan dosyalari lis

#### OPEN KODUNUN PARAMETRELERİ
# 'r' --> Okumak için ac. Dosya yoksa "FileNotFoundError" hatasi alinir(
# 'a' --> Yazmak için ac. Zaten varsa, dosyanın sonuna eklenir. Dosya yo
# 'w' --> Yazmaya için ac. Dosya zaten mevcutsa, üzerine yazılacaktır. D
# 'x' --> Özel oluşturma için aç, dosya zaten mevcutsa başarısız olur.
# 'b' --> ikili modda ac.
# 't' --> Metin dosyasi olarak ac
# '+' --> Güncellemeye acik (okuma ve yazma)

#### -- OKUMA KOMUTLARI -- ####

dosya = open('deneme.txt', 'r', encoding='utf-8') # belirtilen
print(dosya.read()) # acilan dos
print(dosya.read(25)) # 25. satira
print(dosya.readline()) # metnin ilk
```

```

print(dosya.readline(20))
print(dosya.readlines())
print(len(dosya.read()))
print(type(dosya.read()))
dosya.seek(0)
print(dosya.tell())
dosya.close()

### Dosya hakkında bilgi alinabilecek methodlar
d = open('../deneme.txt', 'r', encoding='utf-8')
print('Dosya adi...:', d.name)
print('Dosya kapali mi...:', d.closed)
print('Dosya modu...:', d.mode)
d.close()

with open('myfile.txt', 'r') as file:
    for i in file:
        print(i)

### -- YAZMA KOMUTLARI -- ###

with open('deneme.txt', 'w', encoding='utf-8') as file:
    file.write('Ne olursan ol, yine de gel!')

fruits = ['Banana\n', 'Orange\n', 'Apple\n', 'Cherry\n']
with open('test1.txt', 'w', encoding='utf-8') as f:
    f.writelines(fruits)
    #for i in fruits:
    #    f.write(i)

with open('test1.txt', 'r', encoding='utf-8') as f:
    with open('test2.txt', 'w', encoding='utf-8') as d:
        d.write(f.read())

with open('car.jpg', 'rb') as car:
    with open('newcar.jpg', 'wb') as newcar:
        newcar.write(car.read())

### -- CSV DOSYA KOMUTLARI -- ###

import csv
with open('people.csv', 'r', newline='') as f:
    oku = csv.reader(f, delimiter=',')
    for i in oku:
        print(i)
        print(''.join(i))

veri = [['4', 'Mert', 'Öztürk', '36'],
        ['5', 'Tolga', 'Calis', '32']]
with open('people.csv', 'a', newline='') as a:
    yaz = csv.writer(a)
    yaz.writerow(veri)
    yaz.writerows(veri)

with open('fruit.csv', 'r', encoding='utf-8') as d:
    print(d.read())

with open('people.csv', 'a', encoding='utf-8') as d:
    d.write('\n7, Buse, Celik, 23')

```

```

# mevcut sat
# Tüm satirl
# metnin kar
# metnin tip
# imlecin ko
# imlecin ko
# acilan dos

# Dosya ile
# Dosyanin a
# Dosya açık
# Dosyanin h

# Bu kullani
# Döngüler i

# "w" ile ya
# yazilacak

# sonlarına
# liste form
# "writeline
# her ögeyi

# aktarilmek
# dosyanin i
# burada act

# binary dos
# binary dos
# farkli bir

# csv kullan
# "newline"
# "delimiter
# metni iter
# her satiri
# string ifa

# verileri d

# Dosyaya ye
# önce veril
# satirlari
# tüm satirl

# "csv" modü
# "csv" modü
# alt satira

```