# Built-in Functions

May 23, 2022

### 0.0.1 Built-in Functions (Quraşdırılmış funksiyalar)

Python dili ilə gələn quraşdırılmış funksiyaların bəzilərin indiyə qədər öyrəndik.Bu dərsdə həm keçirilmişləri təkrarlayacaqsınız,həmçinin yeni funksiyalar öyrənəcəksiniz.Dilimizə alternativ olaraq kök və ya quraşdırılmış funksiyalar kimi adlandırdım.Misal olaraq list() str() int() complex() və s göstərmək olar

Quraşdırılmış funksiyaların siyahısını bu https://docs.python.org/3/library/functions.html səhifədə görə bilərsiniz.

[ ]:

all() funksiyası

funksiya bool tipində dəyər ifadə edir. gər ifadələrdən hər biri True isə True,ən az biri False isə False dəyərini verəcək.

[65]:
```
def bütün(iterable):
    for element in iterable:
        if not element:
            return False
    return True
siyahı = [13,1,2,'Python','C++']
```

[66]:
```
bütün(siyahı)
```

[66]: True

Yuxarıda funksiya tərtib etdik funksiyaya dəyər olaraq siyahı tipində ifadələri göndərib sorğuya çəkdik.siyahı daxilində olan bütün elementlər susmaya görə True cavabını verir

[67]:
```
bool(13)
```

[67]: True

[68]:
```
bool('Python')
```

[68]: True

[69]:
```
bool('')
```

[69]: False

[70]:
```
bool(0)
```

[70]: False

Biz bilirikki 0 və boşluqdan savayı bütün simvollar True cavabını verir.Yuxarıdakı funksiyaya ən az biri False dəyərinə uyğun element əlavə edək

[71]:
```
def bütün(iterable):
    for element in iterable:
        if not element:
            return False
    return True
siyahı = [0,1,2,'Python','C++']
bütün(siyahı)
```

[71]: False

Yazdığımız funksiyanın yerinə all() funksiyasından istifadə edərək eyni nəticəni ala bilərik

[72]:
```
siyahı = [0,1,2,'Python','C++']
print(all(siyahı))
siyahı = [13,1,2,'Python','C++']
print(all(siyahı))
```

```
False
True
```

[ ]:

any() funksiyası

funksiya bool tipində dəyər ifadə edir. gər ifadələrdən hər biri False isə False,ən az biri True isə True dəyərini verəcək.

[73]:
```
def hərhansı(iterable):
    for element in iterable:
        if element:
            return True
    return False
siyahı = [0,'',0]
hərhansı(siyahı)
```

[73]: False

[74]:
```
def hərhansı(iterable):
    for element in iterable:
        if element:
```

```
            return True
    return False
siyahı = [0,'',0,'Python']
hərhansı(siyahı)
```

[74]: True

[ ]:

[75]:
```
siyahı = [0,'',0]
print(any(siyahı))
siyahı = [0,'',0,'Python']
print(any(siyahı))
```

```
False
True
```

map() funksiyası

map(function, iterable [, iterable2, iterable3,...iterableN]) –> map object

Funksiya vasitəsilə verilən tiplərindən gələn dəyər və ya dəyərləri,təyin etdiyimiz funksiyaya elementləri tək tək göndərərək,verilən tipinə uyğun məlumatları bizə göndərir

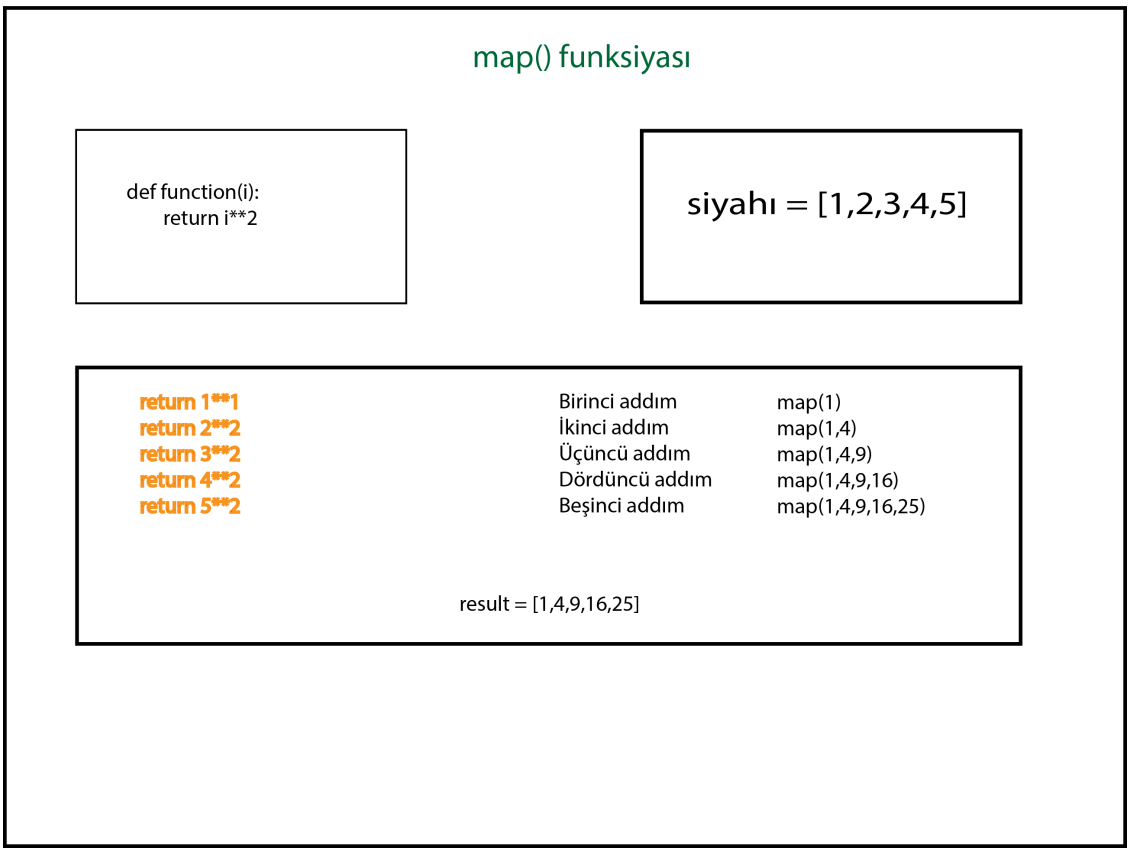
[76]:
```
def func(i):
    return i**2
```

[77]:
```
siyahı = [1,2,3,4,5]
elements = map(func,siyahı)
print(*elements)
print(type(elements))
```

```
1 4 9 16 25
<class 'map'>
```

[78]:
```
from IPython.display import Image
Image("image/map.png")
```

[78]:




[79]:
```
def func(i):
    return i*i
siyahı = [1,2,3,4,5]
elements = map(func,siyahı)
print(list(elements))
```

```
[1, 4, 9, 16, 25]
```

Map with Lambda Expression

[80]:
```
sqrt = lambda x,y: x*x+y*y
```

[81]:
```
sqrt(3,3)
```

[81]:
```
18
```

[82]:
```
square = map(lambda x: x*x,(3,3))
print(list(square))
```

```
[9, 9]
```

[83]:
```
square = map(lambda x: x*x,[1,2,3,4,5,6])
print(list(square))
```

```
[1, 4, 9, 16, 25, 36]
```

[84]:
```
square = map(lambda x,y: x*x+y*y,[1,2,3],[4,5,6])
print(list(square))
```

```
[17, 29, 45]
```

[85]:
```
#və ya
siyahı1 = [1,2,3,4]
siyahı2 = [1,2,3,4]
square = map(lambda x,y: x*x+y*y,siyahı1,siyahı2)
square = [i for i in square]
print(square)
```

```
[2, 8, 18, 32]
```

[86]:
```
print("""
first iteration:  1*1+1*1 = 2
second iteration: 2*2+2*2=8
third iteration:  3*3+3*3=18
fourth iteration: 4*4+4*4=32
""")
```

```
first iteration:  1*1+1*1 = 2
second iteration: 2*2+2*2=8
third iteration:  3*3+3*3=18
fourth iteration: 4*4+4*4=32
```

Qeyd edimki hər iterasiyada(təkrarlama) elementlər qarşılıqlı say nisbətinə uyğun olmalıdır. ks haldı qarşılığı olmayan elementlər nəzərə alınmır və iterasiya sonlandırılır.Yəni siyahıda 10 element varsa ikinci siyahıda isə 11 element olduğu halda birinci siyahının 10cu elementdi iterasiya qarşılığını ikinci siyahıdan alıb təkrarlamanı sonlandıracaq

[87]:
```
siyahı1 = [1,2,3,4]
siyahı2 = [1,2,3,4,5,6]
square = map(lambda x,y: x*x+y*y,siyahı1,siyahı2)
square = [i for i in square]
print(square)
```

```
[2, 8, 18, 32]
```

Yuxarıdakı nümunədə siyahı2 elementlərini artırsaqda siyahı1 də 4 sayda element olduğundan sonuncu element iterasiyanı sonlandırdı.Üstün cəhətlərindən biri isə StopİteratIon xətası verməməsidir.Bu xəta ilə irəliləyən dərslərdə rastlaşacağıq

gər xatırlayırsınızsa dekorativlər dərsində farnheyt və selsi cinsindən temperatur çevirmələri funksiyaları hazırlamışdıq.map funksiyasından istifadə edərək bir necə sayda temperaturu çevirək

[88]:
```
def fahrenheit(celsius):
    return (9/5)*celsius + 32

F = map(fahrenheit,[0,1,12,4,35,18])
print(list(F))
```

```
[32.0, 33.8, 53.6, 39.2, 95.0, 64.4]
```

[ ]:
```

```

dir() funksiyası

funksiya vasitəsilə obyektin tipinə uyğun metodları öyrənə bilirik.

[89]:
```
kortej = (1,2,3,'C++')
```

[90]:
```
dir(kortej)
```

[90]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmul__',
```

```
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__',
'count',
'index']
```

[91]:
```
class Vehicle():
    def __init__(self,mark,engine,color,car):
        self.mark = mark
        self.engine = engine
        self.color = color
        self.car = car
    def __dir__(self):
        return␣
 ↪['__str__','__del__','__len__','__init__','mark','engine','color','car']
    def __len__(self):
        return self.count
    def __str__(self):
        return ("""
        Mark:{}
        Engine:{}
        Color:{}
        Car: {}
        """.format(self.mark,self.engine,self.color,self.car))
    def __del__(self):
        return ('Object deleted')
```

[92]:
```
vehicle = Vehicle('Mercedes','Mak','black',1)
```

[93]:
```
print(vehicle)
```

```

        Mark:Mercedes
        Engine:Mak
        Color:black
        Car: 1

```

[94]:
```
dir(vehicle)
```

[94]:
```
['__del__', '__init__', '__len__', '__str__', 'car', 'color', 'engine', 'mark']
```

[ ]:

[ ]:

enumerate() funksiyası

funksiya vasitəsilə verilən tipi elementləri sırası ilə nömrələndirmək mümkündür.Nömrələməni kortej tipində edərək bizə enumerate dəyər göndərir.

[95]:
```
def example(y):
    sayğac =0
    siyahı = []
    for i in y:
        sayğac+=1
        siyahı.append((sayğac,i))
    return siyahı

example(['Python','Java','C++'])
```

[95]:
```
[(1, 'Python'), (2, 'Java'), (3, 'C++')]
```

Yuxarıda funksiyada qeyd etdiyimiz elementləri nömrələndirdik.Və ilk 1-ədədindən başladı.Yuxarıdakı funksiya əvəzinə enumerate funksiyasını istifadə edərək qısa yoldan nəticəni əldə edə bilərik

[96]:
```
siyahı = ['Python','Java','C++','Perl']
print(list(enumerate(siyahı)))#enumerate dəyərini siyahı tipinə çevirdiyimiz␣
 ↪üçün siyahı tipində məlumat alacağıq
print(type(siyahı))
```

```
[(0, 'Python'), (1, 'Java'), (2, 'C++'), (3, 'Perl')]
<class 'list'>
```

[97]:
```
a = [10, 20, 30, 40]
for i in enumerate(a):
    print(i)
```

```
(0, 10)
(1, 20)
(2, 30)
(3, 40)
```

[98]:
```
dictionary = {1: 'a', 2: 'b', 3: 'c'}
for i in enumerate(dictionary.values()):
    print(i)
```

```
(0, 'a')
(1, 'b')
(2, 'c')
```

gər istəsəniz nömrələməni özünüz də qeyd edə bilərsiniz

[99]:
```
for i, j in enumerate('python', 1):
    print("{} {}".format(i,j))
```

```
1 p
2 y
3 t
4 h
5 o
6 n
```

zip() funksiyası

Funksiya parametr olaraq aldığı elementləri qruplandıraraq dəyərini kortej tipində bizə göndərir.İki verilən tipi elementləri arasında say etibarı ilə bərabərlik ödəndiyi iterasiyaya qədər davam edər.Qruplandıracağı qarşılığı olmadıqda isə iterasiya sonlandırılır.

[100]:
```
def Group(iter1,iter2):
    sayğac = 0
    siyahı = []
    while (sayğac<len(iter1) and sayğac<len(iter2)):
        siyahı.append((iter1[sayğac],iter2[sayğac]))
        sayğac+=1
    return siyahı

Group(['Azerbaijan','Germany','Greek'],['Baki','Berlin','Athena','Tehran'])
```

[100]:
```
[('Azerbaijan', 'Baki'), ('Germany', 'Berlin'), ('Greek', 'Athena')]
```

Yuxarıdakı funksiya vasitəsilə hər iki siyahı daxilindəki elementləri qruplaşdırdıq.Tehran elementinə qarşılıq element olmadığından ifadə nəzərə alınmadı

[101]:
```
var = ['Azerbaijan','Germany','Greek']
var2 = ['Baki','Berlin','Athena']
for i in list(zip(var,var2)):
    print(i)
```

```
('Azerbaijan', 'Baki')
('Germany', 'Berlin')
('Greek', 'Athena')
```

[102]:
```
#və ya
var = ['Azerbaijan','Germany','Greek']
var2 = ['Baki','Berlin','Athena']
var3 = [12,16,18,20,22,24,26]
print(list(zip(var,var2,var3))) #zip() funksiyası üçün verilən tipi␣
 ↪məhdudiyyəti yoxdur
```

```
[('Azerbaijan', 'Baki', 12), ('Germany', 'Berlin', 16), ('Greek', 'Athena', 18)]
```

[103]:
```
dict1 = {'Python':['Opencv','Simplecv','pillow']}
dict2 = {'Library':['MOG','KNN','MSE']}
```

[104]:
```
print(list(zip(dict1.keys(),dict2.keys())))
```

```
[('Python', 'Library')]
```

[105]:
```
print(list(zip(dict1.values(),dict2.values())))
```

```
[(['Opencv', 'Simplecv', 'pillow'], ['MOG', 'KNN', 'MSE'])]
```

[ ]:
```

```

filter() funksiyası

Funksiya bool verilən tipində dəyərləndirir.True və ya False.Həmçinin parametr olaraq funksiya və verilən tipində ifadələri alır.İfadələri funksiyaya göndərərək True elementlərini bizə göstərir.Misal olaraq tək və cüt ədədlərdən ibarət diapazonda rəqəmlər var.Biz süzgəcləndirərək sadəcə cüt ədədləri ekrana çap etdirə bilərik

[106]:
```
siyahı = []
for i in range(20):
    if i%2==0:
        siyahı.append(i)
print(siyahı)
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

[107]:
```
def func(i):
    if i%2 == 0:
        return i
var =range(20)
list(filter(func,var))
```

[107]:
```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

[108]:
```
siyahı = list(range(10))
#print(*(filter(lambda x: x%2==0,siyahı)))
print(list(filter(lambda x: x%2==0,siyahı)))
```

```
[0, 2, 4, 6, 8]
```

[109]:
```
score = range(0,101)

def test(x):
    return x >= 70
a = input('Student:')
b = int(input('Score:'))
result = list(filter(test,score))
if b in result:
    print('{},sizi təbrik edirik.{} keçid balını toplamısınız.'.format(a,b))
```

```
else:
    print('{},Təəssüf edirik!'.format(a))
```

```
Student:Eldar
Score:69
Eldar,Təəssüf edirik!
```

[110]:
```
score = range(0,101)

def test(x):
    return x >= 70
a = input('Student:')
b = int(input('Score:'))
result = list(filter(test,score))
if b in result:
    print('{},sizi təbrik edirik.{} keçid balını toplamısınız.'.format(a,b))
else:
    print('{},Təəssüf edirik!'.format(a))
```

```
Student:Kəmalə
Score:71
Kəmalə,sizi təbrik edirik.71 keçid balını toplamısınız.
```

[ ]:
```

```

reduce funksiyası

Funksiya artıq functools moduluna əlavə olunduğu üçün biz functools modulunu daxil etməliyik(import) ki reduce funksiyasını istifadə edək.

Funksiya verilmiş ədədlər diapazonunu iki-iki qruplandıraraq tətbiq etdiyimiz digər funksiyanın əməliyyatını yerinə yetirir.Sözlə ifadə etmək qəliz olduğu üçün nümunə kodlarla bunu daha yaxşı başa düşəcəksiniz.

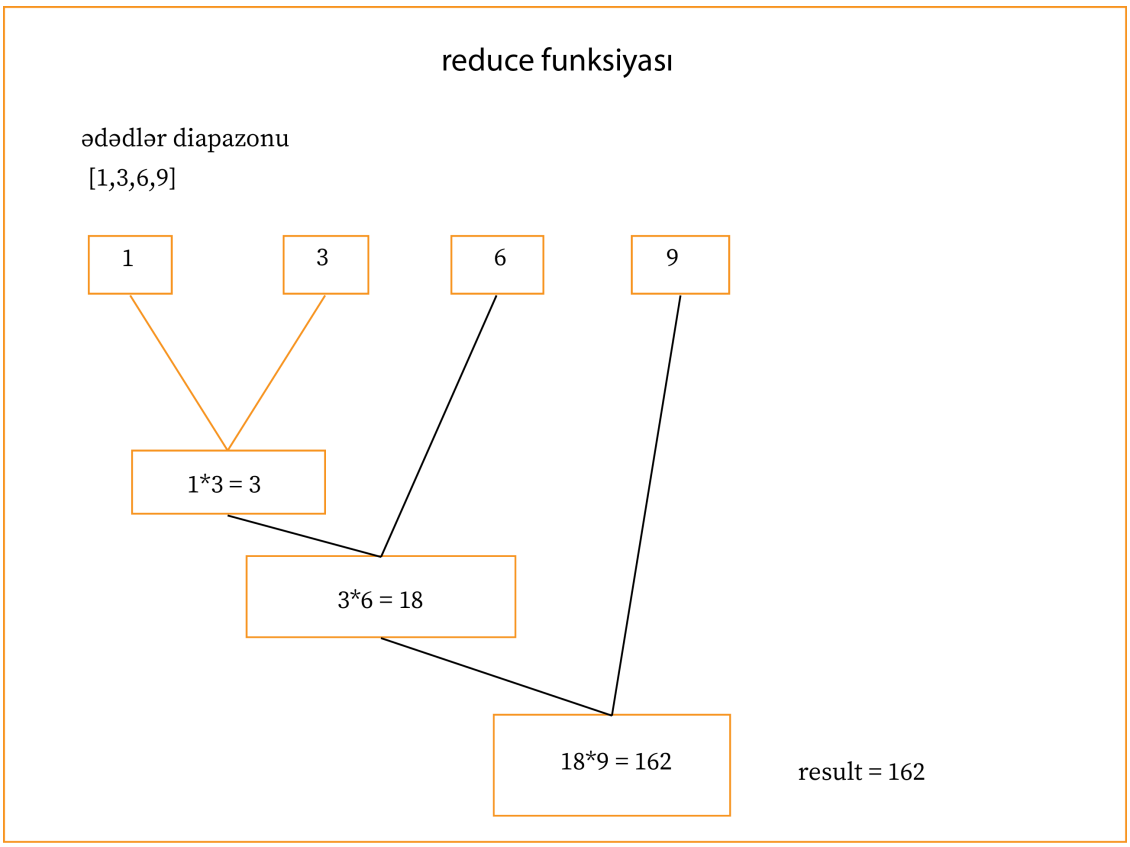
[111]:
```
from functools import reduce #import edirik

def func(a,b):
    return a*b
print(reduce(func,[1,3,6,9]))
```

```
162
```

siyahı diapazonunda 1 və 3 ədədləri funksiyaya göndərilir və alınan nəticə yadda saxlanılır.və siyahımız ilk iterasiyada [3,6,9] diapazonunda olur.Daha sonra 3 və 6 rəqəmlərini funksiyaya göndərilir və alınan cavab siyahıya əlavə olunur.və siyahımız [18,9] formasında olur.Son olaraq 18 və 9 rəqəmləri funksiyaya göndərilir alınan cavab son nəticə olduğu üçün iterasiya sonlandırılır

[112]:
```
from IPython.display import Image
Image('image/reduce.png')
```

[112]:




[113]:
```
from functools import reduce #import edirik

def func(a,b):
    return a+b
print(reduce(func,[10,20,30,40]))
```

```
100
```

Yuxarıdakı funksiyada isə vurma əvəzinə toplama istifadə etdik.Eyni qayda ilə ilk iki element seçilir 10 və 20 funksiyada toplanaraq 30 cavabı növbəti 30 rəqəmi ilə funksiyaya göndərilir və alınan cavab 60 son 40 rəqəmi ilə funksiyaya göndərilərək 100 cavabını əldə edirik.iterasiyada ədəd sayı bitdiyi üçün sonlandırılır.

[ ]:
```

```

len() funksiyası

Funksiyadan daha öncə də istifadə etmişik

[115]:
```
a = input('Adınız:')
print(len(a))
```

```
Adınız:Niyazi
6
```

[ ]:

abs() funksiyası

funksiya ədədin mütləq dəyərini verir

[116]:
```
a = float(input(' dəd yazın:'))
print(abs(a))
```

```
 dəd yazın:34
34.0
```

[117]:
```
a = float(input(' dəd yazın:'))
print(abs(a))
```

```
 dəd yazın:-23
23.0
```

[ ]:

min() və max() funksiyaları

[118]:
```
siyahı = list(range(50))
print('Maksimum dəyər',max(siyahı))
print('Minimum dəyər',max(siyahı))
```

```
Maksimum dəyər 49
Minimum dəyər 49
```

[119]:
```
print(max("Python"))
```

```
y
```

[120]:
```
print(min("Python"))
```

```
P
```

[ ]:

round() funksiyası

Funksiya,həqiqi ədəd tipində ifadələri yuvarlaqlaşdırır.

[121]:
```
print(round(1.42014)) #.5 ə yxın olsada 5 -deyil
print(round(3.50000003))
print(round(8.49))
print(round(8.6000012))
```

```
1
4
8
9
```

[ ]:

divmod() funksiyası

Xatırlayırsınızsa bir ədədin digər ədədə bölünməsindən qalan qalığı alırdıq.Funksiya isə həm qalığı həmdə bölənin nəqədər olduğunu göstərir

[122]:
```
12%4
```

[122]: 0

[123]:
```
12%5
```

[123]: 2

[124]:
```
print(divmod(12,5)) # 5 ədədi 12-də 2 dəfə qalan qalıq isə 2-dir
```

```
(2, 2)
```

pow() funksiyası

Funksiya vasitəsilə qüvvətə yüksəltməni yerinə yetirə bilərik.

[125]:
```
print(pow(2,2))
```

```
4
```

[126]:
```
print(pow(2,3))
```

```
8
```

[127]:
```
def func(a,b):
    return (pow(a,b))
func(2,3)
```

[127]: 8

[ ]:

sum() funksiyası

Funksiya vasitəsilə verilən ədədlərin cəmini əldə edə bilərik

[128]:
```
def Cəm(a):
    return (sum(a))
Cəm([1,2,3,4,5,6,7,8,9,0]) #Siyahı
```

[128]: 45

[129]:
```
def Cəm(a):
    return (sum(a))
Cəm((1,2,3,4,5,6,7,8,9,0)) #Kortej
```

[129]: 45

isinstance() Funksiyası

Funksiya obyektin sorğuya çəkiləcək verilən tipi ilə qarşılaşdırılıb bool tipində dəyər verir

[130]:
```
numbers = [1, 2, 3]
result = isinstance(numbers, list)
print(numbers,'İfadəsi siyahıdırmı?', result)
result = isinstance(numbers, dict)
print(numbers,'İfadəsi lüğətdirmi?', result)
result = isinstance(numbers, (dict, list))
print(numbers,'İfadəsi siyahı və ya lüğətdirmi?', result)
number = 5
result = isinstance(number, list)
print(number,'İfadəsi siyahıdırmı?', result)
result = isinstance(number, int)
print(number,'İfadəsi tam ədəddirmi?', result)
```

```
[1, 2, 3] İfadəsi siyahıdırmı? True
[1, 2, 3] İfadəsi lüğətdirmi? False
[1, 2, 3] İfadəsi siyahı və ya lüğətdirmi? True
5 İfadəsi siyahıdırmı? False
5 İfadəsi tam ədəddirmi? True
```

[ ]:

reversed() funksiyası

[131]:
```
string = 'Python'
```

[132]:
```
string[0]
```

[132]: 'P'

[133]:
```
string[::-1] #tərs çevirdik
```

[133]: 'nohtyP'

[134]:
```
print(list(reversed(string)))
```

```
['n', 'o', 'h', 't', 'y', 'P']
```

[135]:
```
print(*(reversed(string)))
```

```
n o h t y P
```

[136]:
```
siyahı = [1,2,3,4,5]
print(list(reversed(siyahı)))
```

```
[5, 4, 3, 2, 1]
```

[137]:
```
class Çevir():
    atribute = ['a', 'ı', 'o', 'u']
    def __reversed__(self):
        return reversed(self.atribute)
v = Çevir()
print(list(reversed(v)))
```

```
['u', 'o', 'ı', 'a']
```

[ ]:

locals() və globals() Funksiyaları

Funksiya vasitəsilə obyektin local dəyişənlərini öyrənə bilərik

[138]:
```
def f(x,y):
    a = 20
    b = 30
    print(locals())
f(5,10)
```

```
{'b': 30, 'a': 20, 'y': 10, 'x': 5}
```

[ ]:

[139]:
```
a =3
def func(x,y):
    a = 12
    print(locals())
    print('*'*50)

func(6,7)
```

```
{'a': 12, 'y': 7, 'x': 6}
**************************************************
```

[140]:
```
c = 4
def func():
    c = 10
    print(locals())
```

```
    print('*'*40)
    globals()['c'] = c
    print (c)
func()
```

```
{'c': 10}
****************************************
10
```

Funksiyadan xaric global c dəyişənin dəyəri 4-dür.funksiya daxilində local c -dəyişəninin dəyəri isə 10-dur.Təkrar edərək qeyd edimki funksiyadan xaric c-ilə funksiya daxilində olan c dəyişəni tamamilə fərqli dəyişənlərdir.Və öncə local dəyişəni ekrana çap etdirdik.Daha sonra funksiyadan xaric c-dəyişəninin dəyərini local c-dəyişənin dəyəri ilə əvəzlədik

[ ]:

[ ]: