

Operators

May 23, 2022

1 Python Conditions and If statements

Pythonda Şərt operatorları (if else elif)

Hər bir dildə olduğu kimi python dilində də şərt operatorları var.Şərt operatorları verilmiş argumentlərə qarşı şərt qoyaraq kod blokunu təşkil edir.Bura qədər öyrəndiyimiz ümumi girişdən savayı data tipləri mövzusu necə vacibdirsə şərt operatorları da bir o qədər vacib nüans sayılır.Demək olarki şərt operatorlarını öyrəndiyiniz müddətdə kod sətirləri asta-asta mürəkkəbləşəcək və beyninizdə sual yaradacaq bəzi suallar daha da aydınlaşacaq.Şərt operatorları istifadəsi zamanı dil özünəməxsus abzas(indentation) xüsusiyyəti ilə operatora daxil olan ifadələri həmin abzasdan yazılma məcburiyyətini tələb edir.abzas 4 dəfə boşluq (space) düyməsinə və ya bir tab düyməsi qədər məsafədə olub ifadələrin yazılmasını o nöqtədən hesaba alır.if Operatoru həmçinin dildə açar söz olub tətbiq edilməsində, məntiq operatorlarını təşkil edə bilir .Şərt operatorlarını öyrəndiyiniz müddətdə həmçinin əvvəlki mövzuları da təkrarlamış olacaqsınız.Odurki nəzəri hissəni çox uzatmayıb if operatoruna nəzər yetirək

```
[1]: print("""  
  
if argument:  
    kodlar;  
if argument1:  
    kodlar;  
  
""")
```

```
if argument:  
    kodlar;  
if argument1:  
    kodlar;
```

```
[38]: a = 33
      b = 200
      if b > a:
      print("{} {}-ddən böyükdür.".format(b,a))
```

```
File "<ipython-input-38-61b40b9d3259>", line 4
    print("{} {}-ddən böyükdür.".format(b,a))
    ^
```

IndentationError: expected an indented block

Yuxarıdakı nümunədə abzas qorunması təşkil olunmadığından IndentationError: expected an indented block -xətası aldıq

```
[39]: a = 33
      b = 200
      if b > a:
          print("{} {}-dən böyükdür.".format(b,a))
```

200 33-ddən böyükdür.

Formasında ifadə olunur.Nümunələrə baxaq

```
[2]: siyahı = ['Python','Java','C++']
     'Python' in siyahı
```

[2]: True

ifadəsinin yerinə if operatorundan istifadə edərək

```
[3]: siyahı = ['Python','Java','C++']
     if 'Python' in siyahı:
         print('Bəli "Python" ifadəsi lüğət daxilində var!')
```

Bəli "Python" ifadəsi lüğət daxilində var!

Yuxarıdakı nümunədə siyahı tərtib edib Python ifadəsinin siyahı daxilində var olub-olmadığını sorğuya çəkdik. gər Python ifadəsi siyahı daxilində varsa ekrana ifadələr yaz əmrini verdik.

Biz bilirikki verdiyimiz ifadə siyahı daxilində mövcuddur,bunun əksi olaraq sorğuya çəkdiyimiz ifadə əgər olmasa necə!

```
[4]: siyahı = ['Python','Java','C++']
     if 'Rust' in siyahı:
         print('Bəli "Python" ifadəsi lüğət daxilində var!')
```

Yuxarıdakı nümunədə Rust ifadəsini sorğuya çəkdim və bilirikki bu ifadə siyahı daxilində mövcud deyil.Bu ifadələri biz şərti olaraq qeyd edirik,proqramları biz yönəldə bilmərik hər zaman.İfadəni istifadəçidən alsaq istifadəçi toplama işarəsi girərsə ekrana heç bir ifadə çap olunmayaraq proqram

bitəcəkdir. Bunun alternativi else operatorudur. if ifadəsinin əksinə olan məqamlarda dövrəyə girərək daxilində məlumatları çalışdırır.

```
[278]: siyahı = ['Python', 'Java', 'C++', 'Rust']
data = input('Siyahı daxilində axtarış üçün ifadə yazın : ')
if data in siyahı:
    print('*'*20)
    print('Bəli "{}" ifadəsi siyahı daxilində var!'.format(data))
else:
    print('Xeyr "{}" ifadəsi siyahı daxilində yoxdur!'.format(data))
```

Siyahı daxilində axtarış üçün ifadə yazın : Rust

Bəli "Rust" ifadəsi siyahı daxilində var!

```
[279]: siyahı = ['Python', 'Java', 'C++', 'Rust']
data = input('Siyahı daxilində axtarış üçün ifadə yazın : ')
if data in siyahı:
    print('*'*20)
    print('Bəli "{}" ifadəsi siyahı daxilində var!'.format(data))
else:
    print('Xeyr "{}" ifadəsi siyahı daxilində yoxdur!'.format(data))
```

Siyahı daxilində axtarış üçün ifadə yazın : Go

Xeyr "Go" ifadəsi siyahı daxilində yoxdur!

Digər nümunələrə baxaq

```
[7]: num1 = float(input('Number:'))
num2 = float(input('Number:'))
result = None;
if num1>num2:
    result = num1/num2
    print('Cavab:{}'.format(result))
else:
    result = num2/num1;
    print('Cavab:{}'.format(result))
print('Bölmə əməlini yerinə yetirdik')
```

Number:12

Number:5

Cavab:2.4

Bölmə əməlini yerinə yetirdik

```
[8]: num1 = float(input('Number:'))
num2 = float(input('Number:'))
result = None;
```

```

if num1>num2:
    result = num1/num2
    print('Cavab:{}'.format(result))
else:
    result = num2/num1;
    print('Cavab:{}'.format(result))
print('Bölmə əməlini yerinə yetirdik')

```

Number:4

Number:7

Cavab:1.75

Bölmə əməlini yerinə yetirdik

Yuxarıdakı nümunədə istifadəçidən aldığımız iki ədədi müqayisə edib bölmə əməlini yerinə yetirdik

```

[9]: yaş = int(input('Yaşınızı qeyd edin:'));
    limit = 18;
    if yaş >= limit :
        print('Siz yaş məhdudiyyətini keçmisiniz.Filmi izləyə bilərsiniz!')
    else:
        print('Yaşınız 18-dən kiçikdir.Filmi izləmək sizə qadağandır.')

```

Yaşınızı qeyd edin:18

Siz yaş məhdudiyyətini keçmisiniz.Filmi izləyə bilərsiniz!

```

[10]: yaş = int(input('Yaşınızı qeyd edin:'));
    limit = 18;
    if yaş >= limit :
        print('Siz yaş məhdudiyyətini keçmisiniz.Filmi izləyə bilərsiniz!')
    else:
        print('Yaşınız 18-dən kiçikdir.Filmi izləmək sizə qadağandır.')

```

Yaşınızı qeyd edin:16

Yaşınız 18-dən kiçikdir.Filmi izləmək sizə qadağandır.

```

[11]: password = input('Şifrənizi yazın:');
    limit = 6
    if len(password)<limit :
        print('Şifrəniz {} sayda olmalıdır.'.format(limit))

```

Şifrənizi yazın:abc12

Şifrəniz 6 sayda olmalıdır.

```

[12]: password = input('Şifrənizi yazın:');
    limit = 6
    if len(password)<limit :
        print('Şifrəniz {} sayda olmalıdır.'.format(limit))

```

Şifrənizi yazın:abc123

```
[13]: password = input('Şifrənizi yazın:');
limit = 6
if len(password)<limit :
    print('Şifrəniz {} sayda olmalıdır.'.format(limit))
if len(password)==limit:
    print('Şifrəniz {} qəbul olundu.'.format(limit))
    print('Şifrəniz :{}'.format(password))
```

Şifrənizi yazın:abc123
 Şifrəniz 6 qəbul olundu.
 Şifrəniz :abc123

```
[14]: password = input('Şifrənizi yazın:');
limit = 6
if len(password)<limit :
    print('Şifrəniz {} sayda olmalıdır.'.format(limit))
if len(password)==limit:
    print('Şifrəniz "{}" qəbul olundu.'.format(password))
    print('Şifrəniz :{}'.format(password))
```

Şifrənizi yazın:abc1
 Şifrəniz 6 sayda olmalıdır.

elif Operatoru

birdən çox seçimlər qarşısında elif operatorundan istifadə etmək daha məqsəduyğundur

Riyazi əməlləri yerinə yetirən proqram hazırlayaq

```
[15]: print("""
#####
#                                     #
# Sadə Hesablama Proqramı #
#                                     #
#                                     #
#####

'1' Toplama
'2' Çıxma
'3' Vurma
'4' Bölmə
'5' Faiz

""")

#Proqramın tərtibatı

#məliyyatın seçilməsi
```

```

operation = input('Yuxarıdakı əməliyyatlardan birini seçin')

result = None;
if operation == '1':
    num1 = float(input('İlk ədədi yazın:'))
    num2 = float(input('İkinci ədədi yazın:'))
    #Toplama əməli
    result = num1+num2
    print('Cavab : {}'.format(result))

elif operation == '2':
    num1 = float(input('İlk ədədi yazın:'))
    num2 = float(input('İkinci ədədi yazın:'))
    #Çıxma əməli
    result = num1-num2
    print('Cavab : {}'.format(result))

elif operation == '3':
    num1 = float(input('İlk ədədi yazın:'))
    num2 = float(input('İkinci ədədi yazın:'))
    #Vurma əməli
    result = num1*num2
    print('Cavab : {}'.format(result))

elif operation == '4':
    num1 = float(input('İlk ədədi yazın:'))
    num2 = float(input('İkinci ədədi yazın:'))
    #Bölmə əməli
    result = num1/num2
    print('Cavab : {}'.format(result))

elif operation == '5':
    #Faiz əməli
    num1 = float(input('Faizi tapılacaq ədəd:'))
    num2 = float(input('Faiz dərəcəsi:'))
    result = (num1*num2)/100
    print('Cavab : {}'.format(result))

```

```

#####
#                                     #
# Sadə Hesablama Programı #
#                                     #
#                                     #

```

```
#####
```

```
'1' Toplama  
'2' Çıxma  
'3' Vurma  
'4' Bölmə  
'5' Faiz
```

Yuxarıdakı əməliyyatlardan birini seçin5

Faizi tapılacaq ədəd:1000

Faiz dərəcəsi:30

Cavab : 300.0

```
[16]: print("""  
  
#####  
#                                     #  
# Sadə Hesablama Proqramı #  
#                                     #  
#                                     #  
#####  
  
'1' Toplama  
'2' Çıxma  
'3' Vurma  
'4' Bölmə  
'5' Faiz  
  
""")  
  
#Proqramın tərtibatı  
  
#məliyyatın seçilməsi  
  
operation = input('Yuxarıdakı əməliyyatlardan birini seçin')  
  
result = None;  
if operation == '1':  
    num1 = float(input('İlk ədədi yazın:'))  
    num2 = float(input('İkinci ədədi yazın:'))  
    #Toplama əməli  
    result = num1+num2  
    print('Cavab : {}'.format(result))  
  
elif operation == '2':
```

```

num1 = float(input('İlk ədədi yazın:'))
num2 = float(input('İkinci ədədi yazın:'))
#Çıxma əməli
result = num1-num2
print('Cavab : {}'.format(result))

elif operation == '3':
    num1 = float(input('İlk ədədi yazın:'))
    num2 = float(input('İkinci ədədi yazın:'))
    #Vurma əməli
    result = num1*num2
    print('Cavab : {}'.format(result))

elif operation == '4':
    num1 = float(input('İlk ədədi yazın:'))
    num2 = float(input('İkinci ədədi yazın:'))
    #Bölmə əməli
    result = num1/num2
    print('Cavab : {}'.format(result))

elif operation == '5':
    #Faiz əməli
    num1 = float(input('Faizi tapılacaq ədəd:'))
    num2 = float(input('Faiz dərəcəsi:'))
    result = (num1*num2)/100
    print('Cavab : {}'.format(result))

```

```

#####
#                                     #
# Sadə Hesablama Proqramı #
#                                     #
#                                     #
#####

```

```

'1' Toplama
'2' Çıxma
'3' Vurma
'4' Bölmə
'5' Faiz

```

```

Yuxarıdakı əməliyyatlardan birini seçin3
İlk ədədi yazın:12
İkinci ədədi yazın:6
Cavab : 72.0

```



```
[37]: siyahı = [1,2,3,4,5,6,7,8,9,0];
data = int(input('Sorguya çəkəcəyiniz ədədi qeyd edin:'))
if data in siyahı:
    print('{} ədədi siyahı daxilində var'.format(data))
else:
    print('{} ədədi siyahı daxilində yoxdur'.format(data))
```

Sorguya çəkəcəyiniz ədədi qeyd edin:10
10 ədədi siyahı daxilində yoxdur

```
[23]: dictionary = {'kitab':{'Davinchi':{'Author':'Dan Brown','Year':
    ↳'2005','Publisher':'BrightSummaries'},
    'A Life':{'Author':'Gabriel Garcia Marquez','Year':
    ↳'2012','Publisher':'Bloomsbury Publishing'}}}
```

```
[24]: print(dictionary)
```

```
{'kitab': {'Davinchi': {'Author': 'Dan Brown', 'Year': '2005', 'Publisher':
'BrightSummaries'}, 'A Life': {'Author': 'Gabriel Garcia Marquez', 'Year':
'2012', 'Publisher': 'Bloomsbury Publishing'}}}
```

```
[25]: print(dictionary['kitab']['Davinchi']['Year'])
```

2005

```
[26]: print(dictionary.values())
```

```
dict_values([{'Davinchi': {'Author': 'Dan Brown', 'Year': '2005', 'Publisher':
'BrightSummaries'}, 'A Life': {'Author': 'Gabriel Garcia Marquez', 'Year':
'2012', 'Publisher': 'Bloomsbury Publishing'}}])
```

```
[31]: dictionary['kitab']
```

```
[31]: {'Davinchi': {'Author': 'Dan Brown',
    'Year': '2005',
    'Publisher': 'BrightSummaries'},
    'A Life': {'Author': 'Gabriel Garcia Marquez',
    'Year': '2012',
    'Publisher': 'Bloomsbury Publishing'}}
```

```
[36]: book = input('Kitab adını qeyd edin : ')
if book in dictionary['kitab']:
    print("{} kitabı bazamızda var".format(book))
    print('{}'.format(dictionary['kitab'][book]))
else:
    print('Təəssüf edirik,{} kitabı bazamızda yoxdur'.format(book))
```

```
Kitab adını qeyd edin : A Life
"A Life" kitabı bazamızda var
{'Author': 'Gabriel Garcia Marquez', 'Year': '2012', 'Publisher': 'Bloomsbury Publishing'}
```

```
[203]: key = input('lütfən açar sözlərdən istifadə\
edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):')

baza = {'meyvələr':['Alma','Armud','Heyva','Nar','Gilas'],
        'tərəvəzlər':['kartof','soğan','kələm']}

if key == '1':
    print("""
Siz meyvələr bölməsində axtarış edəcəksiniz
""")
    search = input('Dükanda axtarış üçün :')

    if search in baza['meyvələr']:
        print('Bazada {} meyvəsi var'.format(search))
    else:
        print('Bazada {} meyvəsi yoxdur'.format(search))

elif key == '2':
    print("""
Siz tərəvəzlər bölməsində axtarış edəcəksiniz
""")
    search = input('Dükanda axtarış üçün :')

    if search in baza['tərəvəzlər']:
        print('Bazada {} tərəvəzi var'.format(search))
    else:
        print('Təəssüf edirik,Bazada {} tərəvəzi yoxdur'.format(search))
else:
    print('Daxil etdiyiniz açar söz naməlumdur')
```

```
lütfən açar sözlərdən istifadəedin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):1
```

```
Siz meyvələr bölməsində axtarış edəcəksiniz
```

```
Dükanda axtarış üçün :alma
Bazada alma meyvəsi yoxdur
```

yuxarıdakı nümunədə 1-ci əməliyyatı seçib alma ifadəsini qeyd etdik amma meyvənin olmadığı barədə bizə mesaj verdi. Halbuki alma meyvəsi bazada var. və böyük hərflə ifadələri yazdığımız üçün bu problemdən çıxmağın yolu sətir tipi verilənlərin metodlarından istifadə etməkdir.

```
[204]: key = input('lütfən açar sözlərdən istifadə \
edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):')

baza = {'meyvələr':['Alma','Armud','Heyva','Nar','Gilas'],
        'tərəvəzlər':['Kartof','Soğan','Kələm']}

search = input('Dükanda axtarış üçün :')
search = search.capitalize() #aldığımız sətir tipi verilənin ilk hərfini böyük
↪hərfa çeviririk
if key == '1':
    print("""
Siz meyvələr bölməsində axtarış edəcəksiniz
""")

    if search in baza['meyvələr']:
        print('Bazada {} meyvəsi var'.format(search))
    else:
        print('Bazada {} meyvəsi yoxdur'.format(search))

elif key == '2':
    print("""
Siz tərəvəzlər bölməsində axtarış edəcəksiniz
""")

    if search in baza['tərəvəzlər']:
        print('Bazada {} tərəvəzi var'.format(search))
    else:
        print('Təəssüf edirik,Bazada {} tərəvəzi yoxdur'.format(search))
else:
    print('Daxil etdiyiniz açar söz naməlumdur')
```

lütfən açar sözlərdən istifadə edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):1
Dükanda axtarış üçün :alma

Siz meyvələr bölməsində axtarış edəcəksiniz

Bazada Alma meyvəsi var

[]:

[]:

[]:

2 Dövr Operatorları

Həmçinin dövr operatorları iki hissədən ibarətdir. Sayğaç(for) Şərtli(While)

gər Dövrün ilk və son dəyərləri məlumdursa sayğaçlı dövr, əgər təkrar elementlərin sayı öncədən məlum deyilsə şərtli dövrlər adlanır.

```
[41]: siyahı = [1,2,3,4,5,6];  
print(siyahı)  
for i in siyahı:  
    print(i)
```

[1, 2, 3, 4, 5, 6]

1
2
3
4
5
6

Yuxarıdakı nümunədə siyahı tərtib etdik, daha sonra i dəyişəninə siyahı elementlərini hər dövrdə mənimsəyərək print() funksiyası ilə ekrana çap etdirdik

Və 6-ədədini ekrana çap edib dövrü bitirdi. Ümumi qayda olaraq biz qeyd etdikki dövrün ilk və son dəyərləri məlumdursa sayğaçlı(for) dövr adlanır. Bizim yazdığımız nümunədə də siyahı dövrün kökü olub ilk və son elementlərin qiymətləri məlum olduğu üçün (1 və 6) dövrü 6 ədədində sonlandırdı

və ya

```
[42]: string = 'Python';  
for sətir in string:  
    print(sətir)
```

P
y
t
h
o
n

Sətir tipi verilənlərin dəyəri, hər biri simvolu xarakterizə etdiyindən python ifadəsi hərflərə parçalandı.

```
[50]: kortej1 = ('Python', 'C++', 'Java')  
kortej2 = ('Octave', 'Rust', 'Go')  
for i in kortej1:  
    for j in kortej2:  
        print(i,j)
```

Python Octave
Python Rust

```
Python Go
C++ Octave
C++ Rust
C++ Go
Java Octave
Java Rust
Java Go
```

Yuxarıdakı nümunədə iç içə for operatorundan istifadə edərək hər iki kortej daxilində ifadələri qarşı qarşıya çap etdik

```
[56]: kortej = ('Python', 'C++', 'Java')
sayğac = 0
for i in kortej:
    sayğac+=1
    print(sayğac,i)
```

```
1 Python
2 C++
3 Java
```

lüğətlərdən istifadə edərək

```
[64]: dictionary = {'name': ['Namiq', 'Arzu'],
                    'surname': ['Mammadov', 'Garayeva']};

for i in dictionary.values():
    print(i)
```

```
['Namiq', 'Arzu']
['Mammadov', 'Garayeva']
```

```
[65]: dictionary = {'name': ['Namiq', 'Arzu'],
                    'surname': ['Mammadov', 'Garayeva']};

for i in dictionary.keys():
    print(i)
```

```
name
surname
```

```
[68]: siyahı = ['Python', 'C++', 'Javascript', 'Php']
count=0
for i in siyahı:
    print(i)
```

Python
C++
Javascript
Php

```
[76]: listler = []  
kortej = ('Java', 'Python', 'Php')  
for i in kortej:  
    listler.append(i)  
    print(listler)
```

['Java']
['Java', 'Python']
['Java', 'Python', 'Php']

Yuxarıdakı nümunədə hər dövr tətbiq olunduğu zaman kortej daxilindəki elementlər siyahıya əlavə olunur.print() funksiyasını dövr operatoru daxilində yerləşdirdikki,dövr təkrarlandığı müddətdə hansı proseslərin getdiyini görək.

```
[77]: listler = []  
kortej = ('Java', 'Python', 'Php')  
for i in kortej:  
    listler.append(i)  
    print(listler)
```

['Java', 'Python', 'Php']

print() funksiyasını blokdan kənarda qeyd etsək ümumi nəticə ekrana çap olunacaqdır

```
[84]: data = input('Siyahıya əlavə edəcəyiniz elementi yazın:')  
siyahı = []  
for data in data:  
    siyahı.append(data)  
    print(siyahı)
```

Siyahıya əlavə edəcəyiniz elementi yazın:Python
['P', 'y', 't', 'h', 'o', 'n']

```
[85]: siyahı = [[1,2,3,4],[5,6,7,8],['AI', 'ML', 'DL']]  
for v in siyahı:  
    print(v)
```

[1, 2, 3, 4]
[5, 6, 7, 8]
['AI', 'ML', 'DL']

Yuxarıdakı nümunədə əgər hər indeksdəki bütün elementləri almaq istəsək

```
[90]: siyahı = [[1,2,3,4],[5,6,7,8],['AI', 'ML', 'DL', 'SP']]  
for a,b,c,d in siyahı:
```

```
print(a,b,c,d)
```

```
1 2 3 4  
5 6 7 8  
AI ML DL SP
```

formada yazacağım

[91]:

```
[98]: kortej = (1,2,3,4,5)  
summa = 0  
for i in kortej:  
    summa+=i  
    print('Cavab:',summa)
```

```
Cavab: 1  
Cavab: 3  
Cavab: 6  
Cavab: 10  
Cavab: 15
```

```
[110]: kortej = (1,2,3,4,5)  
summa = 0  
for i in kortej:  
    summa+=i  
    print('Cavab:',summa)
```

Cavab: 15

Yuxarıdakı nümunədə siyahı daxilindəki ədədlərin hasilini əldə etmək istədik amma 0 -cavabını aldım.Təbiki dövrü sayğac sıfırdan başladığı üçün 0 ın digər ədədlərə hasilini bizə sıfır cavabını verəcəkdir.Bunun alternativ yolu,sayğacı 1dən başlamaqdır.

```
[112]: s = 1  
for i in [1,2,3,4,5,6,7,8,9]:  
    s *= i  
    print('Cavab:',s)
```

Cavab: 362880

```
[118]: siyahı = set(['Python','Java','C++','Go'])  
for j in siyahı:  
    print(j)
```

```
Java  
Python
```

Go
C++

```
[123]: siyahı = ['Python', 'Java', 'C++', 'Go']
       dəst = set()
       for j in siyahı:
           dəst.add(j)
           print(dəst)
       print('-'*20)
       print(dəst)
       print(type(dəst))
```

```
{'Python'}
{'Java', 'Python'}
{'Java', 'Python', 'C++'}
{'Java', 'Python', 'Go', 'C++'}
-----
{'Java', 'Python', 'Go', 'C++'}
<class 'set'>
```

```
[141]: key = ['name', 'surname', 'email', 'phone']
       value = ['Ramin', 'Aliyev', 'example@gmail.com', '+994500']
       siyahı = []
       for i,j in zip(key,value):
           siyahı.append(i+' '+j)
       print(siyahı)
```

```
['name Ramin', 'surname Aliyev', 'email example@gmail.com', 'phone +994500']
```

```
[151]: siyahı = [[1,2],[3,4],[5,6]]

       for i in siyahı:
           print(i)
       print('*'*20)
       print(siyahı[0])
       print(siyahı[1])
       print(siyahı[2])
```

```
[1, 2]
[3, 4]
[5, 6]
*****
[1, 2]
[3, 4]
[5, 6]
```

```
[152]: siyahı = [[1,2],[3,4],[5,6]]
```



```

for i,j in siyahı:
    print(i,j)
print('*'*20)
print(siyahı[0])
print(siyahı[1])
print(siyahı[2])

```

```

1 2
3 4
5 6
*****
[1, 2]
[3, 4]
[5, 6]

```

```

[158]: dictionary = {'lang':['Python','C++','Javascript'],
                    'library':['Opencv','Pandas','librosa'],
                    'ai':['ML','DL']}

```

```

[159]: for i in dictionary:
        print(i)

```

```

lang
library
ai

```

Yuxarıdakı nümunədə lüğət daxilindən sadəcə açar sözləri əldə etdik.amma dəyərləri əldə edə bilmədik.

```

[160]: dictionary = {'lang':['Python','C++','Javascript'],
                    'library':['Opencv','Pandas','librosa'],
                    'ai':['ML','DL']}
for i in dictionary.values():
    print(i)

```

```

['Python', 'C++', 'Javascript']
['Opencv', 'Pandas', 'librosa']
['ML', 'DL']

```

Dəyərləri yuxarıdakı nümunədə əldə etdik.Bütün elementləri əldə etmək üçün isə həm keys həm values elementlərini iki dəyişən ilə əvəzləsək

```

[161]: dictionary = {'lang':['Python','C++','Javascript'],
                    'library':['Opencv','Pandas','librosa'],
                    'ai':['ML','DL']}
for i,j in dictionary.items():
    print(i,j)

```

```

lang ['Python', 'C++', 'Javascript']

```

```
library ['Opencv', 'Pandas', 'librosa']
ai ['ML', 'DL']
```

```
[218]: dictionary = {'lang':['Python','C++','Javascript'],
                    'library':['Opencv','Pandas','librosa'],
                    'ai':['ML','DL']}
for i,j in dictionary.items():
    print('Açar sözlər: {} | Dəyərlər: {}'.format(i,j))
```

```
Açar sözlər: lang | Dəyərlər: ['Python', 'C++', 'Javascript']
Açar sözlər: library | Dəyərlər: ['Opencv', 'Pandas', 'librosa']
Açar sözlər: ai | Dəyərlər: ['ML', 'DL']
```

```
[216]: siyahı = ['Nərmin','Ziya','Kəmalə','İlahə'];
count = 0
for i in siyahı:
    count +=1
    print(count,i)
```

```
1 Nərmin
2 Ziya
3 Kəmalə
4 İlahə
```

```
[ ]:
```

range() funksiyası

built-in function - Python dilində kök funksiyalardan olub,ədədlər diapazonu təşkil edir.Bu ədədlər bizim verdiyimiz elementlər olub,range() funksiyası elementlərin diapazonunu təşkil edir.range() funksiyası həmçinin bir siyahı tipində ədədlərin diapazonudur.Funksiya 3 argument alaraq,range(start,stop,step) formuluna uyğun davranır.

```
[219]: print(range(0,20))
```

```
range(0, 20)
```

Funksiyanı istifadə etmək istədik amma ekrana heç bir nəticə çap olunmadı,bunun üçün ulduz(*) işarəsindən istifadə edərək

```
[220]: print(*range(0,20))
```

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

ekrana elementləri çap edə bilərik

Həmçinin elementləri dəyişənə əlavə edə bilərik

```
[222]: var = range(0,10);  
print(list(var))
```

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

```
[223]: for i in var:  
        print(i)
```

0
1
2
3
4
5
6
7
8
9

```
[233]: siyahı = ['dödlər']  
siyahı1 = [*range(1,4)]  
dictionary = {}  
  
dictionary = dictionary.fromkeys(siyahı,siyahı1)  
print(dictionary)
```

{'dödlər': [1, 2, 3]}

```
[240]: var = []  
for i in range(0,6):  
    var.append(i)  
    print(var)  
print('-'*30);  
print(var)
```

[0]
[0, 1]
[0, 1, 2]
[0, 1, 2, 3]
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4, 5]

[0, 1, 2, 3, 4, 5]

```
[244]: for num in range(1,10):  
        print('python'*num)
```

python
pythonpython
pythonpythonpython
pythonpythonpythonpython
pythonpythonpythonpythonpython
pythonpythonpythonpythonpythonpython
pythonpythonpythonpythonpythonpythonpython
pythonpythonpythonpythonpythonpythonpythonpython
pythonpythonpythonpythonpythonpythonpythonpythonpython

Cüt ədədlərin təyini

```
[273]: count = 0
      cüt = []
      for i in range(0,50):
          if i % 2 ==0 :
              count+=2
              cüt.append(count)

      print(cüt)
```

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50]

gər funksiya daxilində başlanğıc dəyəri qeyd etməsək funksiya başlanğıc dəyəri sıfırdan sayacaqdır

```
[275]: print(*range(10))
```

0 1 2 3 4 5 6 7 8 9

```
[277]: var = int(input('Aralıq qeyd edin:'))
      lists = []
      for i in range(var):
          lists.append(i)
      print(lists)
```

Aralıq qeyd edin:30

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]

Elementləri neqativləmə metodu

```
[282]: print(*range(0,10,-1))
```

Yuxarıdakı nümunədə ədədləri tərsdən yazmağa çalışdıq amma nəticə olmadı. Bunun üçün funksiya daxilində başlanğıc elementi 10 son elementini isə 0 yazmalıyıq

```
[283]: print(*range(10,0,-1))
```

10 9 8 7 6 5 4 3 2 1

ulduz işarəsindən savayı list funksiyasından da istifadə edib siyahı daxilində göstərə bilərik

```
[284]: siyahı = list(range(15,1,-1))
```

```
[285]: print(siyahı)
```

[15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2]

addım sayı

```
[286]: print(*range(0,10,2))
```

0 2 4 6 8

```
[287]: num = [];  
for i in range(0,50,2):  
    num.append(i)  
print(num)
```

[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]

Yuxarıdakı nümunədə sıfırdan əlliyə qədər bütün ədədlər və iki-iki addımlayaraq diapazonu tərtib etdik.

```
[289]: print(list(range(0,20,3)))# sıfırdan 20-yə qədər və üç-üç addımlayaraq  
↪ ədədləri əldə et
```

[0, 3, 6, 9, 12, 15, 18]

```
[1]: count=0  
for i in range(0,10):  
    count+=i;  
print(count)
```

45

Yuxarıdakı nümunədə sıfırdan 10-a qədər siyahı tərtib edib, count dəyişəninə dövr təkrarlandıqca mənimsəyib topladıq. Və nəticədə cavab olaraq 45 ədədi ekrana çap olundu. Deməli range(0,10) aralığında olan ədədlərin cəmi 45-ə bərabərdir.

List comprehension

python dilində siyahı anlayışı, siyahı tərtib etmək üçün daha qısa metod təklif edir.

Aşağıdakı nümunədə biz range funksiyası ilə tərtib etdiyimiz ədədləri siyahıya daxil edək

```
[2]: siyahı = []  
for i in range(10):  
    siyahı.append(i)  
print(siyahı)
```

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

[]: Yuxarıdakı nümunəyə ekvivalent digər nümunə isə comprehension metodudur.

```
[295]: siyahı = [i for i in range(10)]  
print(siyahı)
```

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

Nəticə eynidirmi!Bəli,daha qısa metod ilə ilk nümunəni yazdıq.Yeni metoda uyğunlaşmağımız biraz zaman ala bilər.Alternativ yolu bol bol nümunələr hazırlamaqdır

İndi gəlin siyahı daxilindəki elementləri 2 -yə vuraq

```
[299]: siyahı = []  
for i in range(10):  
    siyahı.append(i*2)  
  
print(siyahı)
```

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

Daha qısa yoldan isə

```
[301]: siyahı = [i*2 for i in range(10)] # i ifadəsindən sonra əməliyyatı yazırıq  
print(siyahı)
```

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

```
[1]: var = [i**2 for i in [1,2,3,4,5,6,7,8,9]]  
print(var)
```

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

Yuxarıdakı nümunədə 1-dən 10-a qədər ədədlər tərtib edib siyahı daxilində hər bir ədədi qüvvətə yüksəltдик

```
[21]: matrix = []  
for i in range(4):  
    matrix.append([])  
    for j in range(4):  
        matrix[i].append(j)  
print(matrix)
```

```
[[0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3]]
```

Yuxarıdakı nümunədə matrix adlı siyahı tərtib etdik, daha sonra 0-dan 4-ə qədər aralıqda ədədləri tərtib edib i-dəyişəninə mənimsətdik. Növbəti dövrün tətbiqində j-dəyişəninə ədəd siyahı mənimsədərək matrix siyahısına append metodu vasitəsilə hər biri 4-ə qədər ədədlər siyahısını əlavə etdik

```
[ ]:
```

2.0.1 Dict comprehension

Siyahlara tətbiq etdiyimiz “siyahı anlayışı” metodunu, lüğətlərə də tətbiq edə bilərik.

```
[ ]:
```

```
[67]: dictionary = {}  
string = 'Python'  
for i in range(1, 6):  
    dictionary[i] = string  
print(dictionary)
```

```
{1: 'Python', 2: 'Python', 3: 'Python', 4: 'Python', 5: 'Python'}
```

Yuxarıdakı ifadənin yerinə

```
[68]: dictionary = {}  
string = 'Python'  
dictionary = {i: string for i in range(1,6)}  
print(dictionary)
```

```
{1: 'Python', 2: 'Python', 3: 'Python', 4: 'Python', 5: 'Python'}
```

kimi də yazıla bilər

```
[82]: dictionary1={i:i for i in range(1, 11)};print(dictionary1)
```

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

```
[ ]:
```

```
[99]: siyahı = ['Python', 'Java', 'Python', 'C++', 'Pascal', 'Fortran', 'Java']  
for i in siyahı:  
    siyahı.count(i)  
    print(siyahı.count(i))
```

```
2  
2  
2  
1  
1
```

1
2

Yuxarıdakı nümunədə siyahı tərtib edib daxilinə elementlərdən bir neçəsini təkrar olunması şərtilə əlavə etdik.Daha sonra hər dövrü siyahı daxilində elementlərin sayını count metodu ilə hesabladıq.elementlərin sayı bitdiyi üçün dövrə tamamlandı

daha qısa yoldan lüğət anlayışı metodu ilə bunu tərtib edə bilərik

```
[97]: dict1 = {i:siyahı.count(i) for i in siyahı}
```

```
[98]: print(dict1)
```

```
{'Python': 2, 'Java': 2, 'C++': 1, 'Pascal': 1, 'Fortran': 1}
```

```
[ ]:
```

2.0.2 Note!

Qeyd edimki kortej anlayışını (tuple comprehension) tətbiq edə bilmirik.Mötərizə kortejlərə mənsub xüsusiyyət olsada,bu qayda kortej anlayışına tətbiq oluna bilinmir.Çünki həmin obyekt(kortej) generasiya mexanizminə aid olur.İrəliləyən dərslərdə silsilələr və törəmə-generasiya haqqında ətraflı öyrənəcəksiniz.İstərsəniz nümunələrdə bunu göstərək

```
[4]: kortej = (i for i in range(10))
```

```
[7]: print(type(kortej))
```

```
<class 'generator'>
```

```
[8]: print(kortej)
```

```
<generator object <genexpr> at 0x0000020A6533ECA8>
```

Gördüyünüz kimi elementləri çap etdirə bilmədik.Bizə generator olduğuna dair məlumat verildi.Siz bunu indi düşünməyin.Qeyd etdiyim kimi irəlidə bun ətraflı öyrənəcəksiniz

2.0.3 While Operatoru

Operator for operatorundan fərqlənərək şərtin doğru və ya false olduğu ,kod bloku daxilində son sətirin dövrü kəsmədiyi müddətcə başa dönüb təkrarlanır

İlk başlarda operatoru for operatoru ilə eyniləşdirə bilərsiniz.Lakin nümunələr yazıldıqca hər iki operatorun harda və necə kodlara yararlı olduğuna dair beyninizdə suallar qalmayacaq.

Operatorndan həm dövrü öncədən bilmədiyimiz təqdirdə istifadə edə ,həmçinin məlum bir şərt varsa operatorndan istifadə edə bilərik

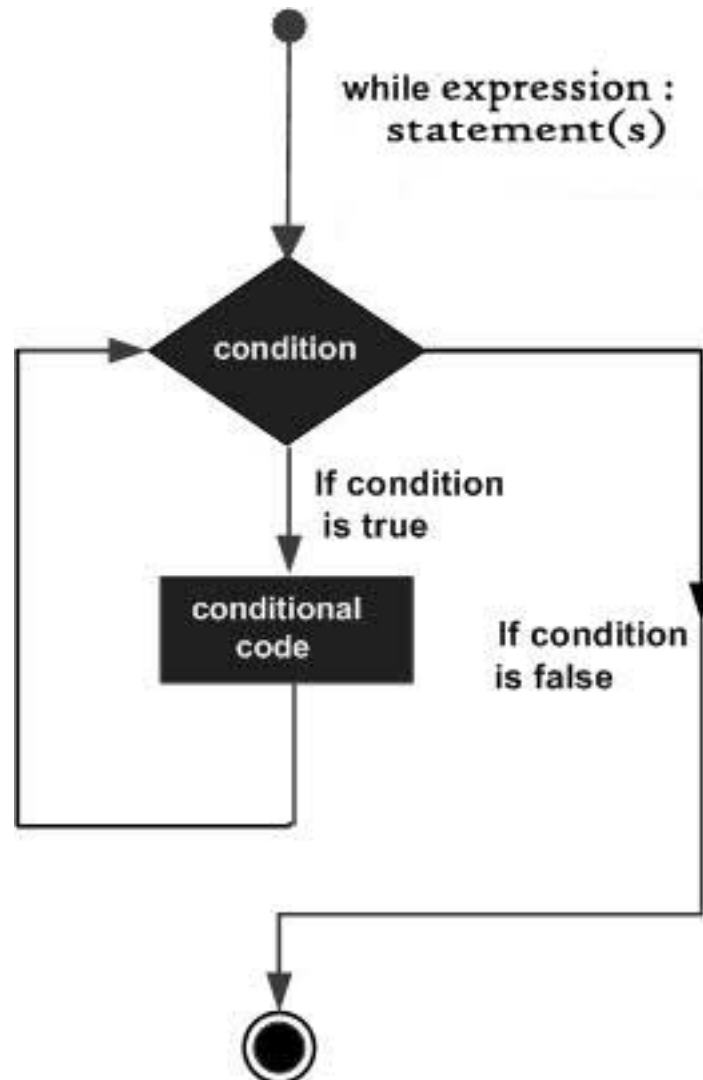
misal olaraq n ədədi 1-ə bərabər olduğu halda n==1 ifadəsi həmişə True cavabını verəcək.Bu şərt məlum olduğu üçün dövrü təkrarlasaq,sonsuz dövr təkrarlanması olduğundan sadə ədədlər üzərində proqramın çökməsinə səbəb olar.İkinci bir hal isə əgər ifadəmiz həmişə doğru olduğu halda sonsuz

dövrü dayandırmaq üçün dövr bloku daxilində yazılmış kod nümunələrindən bilavasitə istifadə olunur. Kod nümunəsində dövrü sonlandırmaq, digər kod nümunələrinə keçid etmək və növbəti şərtlər ilə dövrü dayandırmaq mümkündür.

Aşağıdakı while operatoruna dair diaqram göstərilmişdir.

```
[290]: from IPython.display import Image  
Image("image/loop.jpg")
```

[290]:



```
[119]: print("""  
  
i = 0 #i-dəyişəni hər zaman 0-dır  
  
while (i==0):  
    print(i)
```

```
""")
```

```
i = 0  #i-dəyişəni hər zaman 0-dır
```

```
while (i==0):  
    print(i)
```

```
[103]: i = 0  
       if (i==0):print(True)  
       else: print(False)
```

True

Yuxarıdakı kod nümunəsi hər zaman doğru olduğu üçün sonsuz dövr təkrarlandığından, proqramın çökməsinə səbəb ola bilər

```
[106]: i = 0  #i-dəyişəni hər zaman 0-dır  
  
       while (i==True):  
           print("Python")
```

```
[107]: 0==True
```

```
[107]: False
```

Yuxarıdakı nümunədə isə i-nin dəyəri 0 olduğu üçün True cavabına bərabər deyil. Biz bu dəyəri boolean verilən tipi bəhsindən bilirik. Və şərtimiz doğru olmadığı üçün print() funksiyası öz işini yerinə yetirə bilmədi

gər

```
[115]: i = 0  
       while i < 10:  
           print(i)  
           i += 1  #i-nin dəyərini hər dövrdə bir vahid artırırıq
```

```
0  
1  
2  
3  
4  
5  
6
```

7
8
9

kim yazsaq *i* dəyişəninin dəyəri hər dövrdə bir vahid artıb o zamana qədərki 9 olub 10 ədədinnən kiçikdir şərti doğru olub sonlanır.Qeyd edimki `print()` funksiyasının istifadə olunduğu yerə diqqət edin. gər funksiyanı ən sonda qeyd etsəydik artıq ekrana sayım 0-dan deyil birdən başlayacağı çap olunacaq.ilk öncə şərtə baxılır,*i*=0 10-dan kiçikdirmi bəli,keçir `print()` funksiyasına və dəyəri 0-ı çap edir və son kodda yazılan ifadədə sıfırın üzərinə bir əlavə ed(*i* = 0 + 1)ərək başa dönür.*i* artıq bir olduğunu bilirik və 10-dan kiçikdirmi!Bəli,şərtimiz doğrudur keçirik `print()` funksiyasına,*i*-nin dəyəri bir olduğu üçün ekrana 0-dan sonra bir çap olunur.Və son kod nümunəsində *i* = 1 + 1 ifadəsi hesablanaraq dövr başa keçərək yenidən təkrarlanır.

```
[116]: i = 0
while i < 10:
    i = i + 1 #i-nin dəyərini hər dövrdə bir vahid artırırıq
    print(i)
```

1
2
3
4
5
6
7
8
9
10

Yuxarıdakı nümunənin əksi olaraq `print()` funksiyasını sonda qeyd etdiyimiz üçün *i*-nin dəyəri öncə bir olduğu üçün ekrana sayım birdən başlayaraq çap olundu

və ya

```
[117]: i = 0
while i < 10:
    i = i + 1 #i-nin dəyərini hər dövrdə bir vahid artırırıq
    print(i)
```

10

`print()` funksiyasını sonda qeyd etsək,dövr daima təkrarlanaraq ən son *i*=9 olaraq üzərinə bir əlavə olunub dövr sonlandırılır.Ona görə ekrana ən son nəticə 10 ifadəsi çap olunur

```
[133]: i = 0
lists = []
while(i<10):
    print('Hello World!')
    lists.append(i)
    i+=1
```

```
print("*"*30)
print(lists)
```

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

```
*****
```

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

Yuxarıdakı ifadədə sıfır daxil olaqla i-nin dəyərini 10-a qədər artırdıq və 10- olduğu müddətdə $10 < 10$ dan kiçikdir ifadəsi ödənilmədiyindən dövr təkrarlanmayaraq ekrana 10 dəfə Hello World! ifadəsini çap etdi

Kod blokunda list tərtib etməyimdə məqsədim sizin i-nin dəyərini hər dövrdə artırılmış qiymətini görməyiniz üçündür

```
[ ]:
```

```
[157]: siyahı = [*range(6)] #6-ə qədər başlanğıc dəyəri 0-olan siyahı elementləri
      ↪tərtib etdik
```

```
[158]: print(siyahı)
      print(len(siyahı))
```

```
[0, 1, 2, 3, 4, 5]
```

```
6
```

while operatoru ilə siyahı daxilindəki elementləri alaq

```
[159]: index = 0
      while (index<len(siyahı)):
          print('Index:',index,'Element:',siyahı[index])
          index+=1
```

```
Index: 0 Element: 0
Index: 1 Element: 1
Index: 2 Element: 2
Index: 3 Element: 3
Index: 4 Element: 4
Index: 5 Element: 5
```

Yuxarıdakı nümunə sizə qarışıq gəlməsin.Çünki buna dair nümunələr öncəki bəhslərdə göstərmişdik.Kodlarımızı izah edək

operator daxilində tərtib etdiyimiz şərtə index sayı siyahı elementləri sayından kiçik olduğu müddətcə kodlarımızın çalışmasını tərtib etdik.

[]:

```
[160]: print("""Index:',0,'Element:',siyahı[0]""") #index= 0 olduğu üçün
print("""Index:',1,'Element:',siyahı[1]""") #index= 1 olduğu üçün
print("""Index:',2,'Element:',siyahı[2]""") #index= 2 olduğu üçün
print("""Index:',3,'Element:',siyahı[3]""") #index= 3 olduğu üçün
print("""Index:',4,'Element:',siyahı[4]""") #index= 4 olduğu üçün
print("""Index:',5,'Element:',siyahı[5]""") #index= 5 olduğu üçün
```

```
'Index:',0,'Element:',siyahı[0]
'Index:',1,'Element:',siyahı[1]
'Index:',2,'Element:',siyahı[2]
'Index:',3,'Element:',siyahı[3]
'Index:',4,'Element:',siyahı[4]
'Index:',5,'Element:',siyahı[5]
```

və son olaraq index = 5+1 <6 şərti ödənilmədiyindən dövr sonlanır

[]:

gər xatırlayırsınızsa əvvəlki bəhslərdə meyvə və tərəvəz bazasında axtarış proqramı yazmışdıq.açar sözü daxil etdikdə bazada olub-olmadığını sorğuya çəkib daha sonra proqram bitirdi.

```
[173]: while (True):
    key = input('lütfən açar sözlərdən istifadə \
edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):')

    baza = {'meyvələr':['Alma','Armud','Heyva','Nar','Gilas'],
            'tərəvəzlər':['Kartof','Soğan','Kələm']}

    search = input('Dükanda axtarış üçün :')
    search = search.capitalize() #aldığımız sətir tipi verilənin ilk hərfini
    ↪böyük hərflə çeviririk
    if key == '1':
        print("""
        Siz meyvələr bölməsində axtarış edəcəksiniz
        """)

        if search in baza['meyvələr']:
            print('Bazada {} meyvəsi var'.format(search))
        else:
            print('Bazada {} meyvəsi yoxdur'.format(search))
```

```

elif key == '2':
    print("""
Siz tərəvəzlər bölməsində axtarış edəcəksiniz
    """)

    if search in baza['tərəvəzlər']:
        print('Bazada {} tərəvəzi var'.format(search))
    else:
        print('Təəssüf edirik,Bazada {} tərəvəzi yoxdur'.format(search))

else:
    print('Daxil etdiyiniz açar söz naməlumdur')

```

lutfən açar sözlərdən istifadə edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):1

Dükanda axtarış üçün :alma

Siz meyvələr bölməsində axtarış edəcəksiniz

Bazada Alma meyvəsi var

lutfən açar sözlərdən istifadə edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):2

Dükanda axtarış üçün :soğan

Siz tərəvəzlər bölməsində axtarış edəcəksiniz

Bazada Soğan tərəvəzi var

```

-----
KeyboardInterrupt                                Traceback (most recent call last)
~/.local/lib/python3.7/site-packages/ipykernel/kernelbase.py in _input_request(self, prompt, ident, parent, password)
-> _input_request(self, prompt, ident, parent, password)
    883         try:
--> 884             ident, reply = self.session.recv(self.stdin_socket, 0)
    885         except Exception:

~/.local/lib/python3.7/site-packages/jupyter_client/session.py in recv(self, socket, mode, content, copy)
    802         try:
--> 803             msg_list = socket.recv_multipart(mode, copy=copy)
    804         except zmq.ZMQError as e:

~/.local/lib/python3.7/site-packages/zmq/sugar/socket.py in recv_multipart(self, flags, copy, track)
    474         """

```

```

--> 475         parts = [self.recv(flags, copy=copy, track=track)]
      476         # have first part already, only loop while more to receive

zmq/backend/cython/socket.pyx in zmq.backend.cython.socket.Socket.recv()

zmq/backend/cython/socket.pyx in zmq.backend.cython.socket.Socket.recv()

zmq/backend/cython/socket.pyx in zmq.backend.cython.socket._recv_copy()

~/local/lib/python3.7/site-packages/zmq/backend/cython/checkrc.pxd in zmq.
↳ backend.cython.checkrc._check_rc()

KeyboardInterrupt:

During handling of the above exception, another exception occurred:

KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-173-81199843056e> in <module>
      1 while (True):
      2     key = input('lütfən açar sözlərdən istifadə \
----> 3     edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):')
      4
      5

~/local/lib/python3.7/site-packages/ipykernel/kernelbase.py in raw_input(self,
↳ prompt)
      857         self._parent_ident,
      858         self._parent_header,
--> 859         password=False,
      860     )
      861

~/local/lib/python3.7/site-packages/ipykernel/kernelbase.py in
↳ _input_request(self, prompt, ident, parent, password)
      887         except KeyboardInterrupt:
      888             # re-raise KeyboardInterrupt, to truncate traceback
--> 889             raise KeyboardInterrupt
      890         else:
      891             break

KeyboardInterrupt:

```

Yuxarıda operator daxilində hər zaman True şərtini yazaraq proqramın davamlı çalışmasını təmin etdik. Amma sonda proqramdan çıxış üçün parametr əlavə etmədiyimizdən dolayı proqramı interrupt düyməsini sıxaraq çökdürdük. Proqramdan çıxış üçün irəliləyən bəhslərdə ətraflı keçəcəyimiz sys modulundan istifadə edərək təmin edə bilərik.

```
[170]: import sys
while (True):
    key = input('lütfən açar sözlərdən istifadə \
edin(1-meyvələr bölməsi,2-tərəvəzlər bölməsi):')

    baza = {'meyvələr':['Alma','Armud','Heyva','Nar','Gilas'],
            'tərəvəzlər':['Kartof','Soğan','Kələm']}

    search = input('Dükanda axtarış üçün :')
    search = search.capitalize() #aldığımız sətir tipi verilənin ilk hərfini
    ↪böyük hərfə çeviririk
    if key == '1':
        print("""
        Siz meyvələr bölməsində axtarış edəcəksiniz
        """)

        if search in baza['meyvələr']:
            print('Bazada {} meyvəsi var'.format(search))
        else:
            print('Bazada {} meyvəsi yoxdur'.format(search))

    elif key == '2':
        print("""
        Siz tərəvəzlər bölməsində axtarış edəcəksiniz
        """)

        if search in baza['tərəvəzlər']:
            print('Bazada {} tərəvəzi var'.format(search))
        else:
            print('Təəssüf edirik,Bazada {} tərəvəzi yoxdur'.format(search))
    elif key == 'q':
        sys.exit(0)
    else:
        print('Daxil etdiyiniz açar söz naməlumdur')
```

```
lütfən açar sözlərdən istifadə      edin(1-meyvələr bölməsi,2-tərəvəzlər
bölməsi):1
```

```
Dükanda axtarış üçün :alma
```

```
Siz meyvələr bölməsində axtarış edəcəksiniz
```

```
Bazada Alma meyvəsi var
```

```
lütfən açar sözlərdən istifadə      edin(1-meyvələr bölməsi,2-tərəvəzlər
bölməsi):2
```

```
Dükanda axtarış üçün :kartof
```


Siz tərəvəzlər bölməsində axtarış edəcəksiniz

```
Bazada Kartof tərəvəzi var
lütfn açar sözlərdən istifadə edin(1-meyvələr bölməsi,2-tərəvəzlər
bölməsi):2
Dükanda axtarış üçün :soğan
```

Siz tərəvəzlər bölməsində axtarış edəcəksiniz

```
Bazada Soğan tərəvəzi var
lütfn açar sözlərdən istifadə edin(1-meyvələr bölməsi,2-tərəvəzlər
bölməsi):q
Dükanda axtarış üçün :q
```

An exception has occurred, use %tb to see the full traceback.

SystemExit: 0

2.0.4 while operatoru ilə fibonacci

Tərif əsasən, ilk iki Fibonaççi ədədləri 0 və 1-dir. Sonra gələn ədəd özündən əvvəlki ilk iki qonşusunun cəminə bərabərdir. Bəzi mənbələrdə sıranın ilk ədədi 0 yox, 1 götürülür.

```
[9]: a,b = 0,1 #başlanğıc üçün iki ədəd təyin edirik
      fibb = []
      while (a<22):
          fibb.append(a)
          a,b = b,a+b
      print(fibb)
```

[0, 1, 1, 2, 3, 5, 8, 13, 21]

[]:

[]:

Break , Continue və Pass ifadələri

```
[22]: siyahı = [*range(11)]
      for i in siyahı:
          if (i == 5):
              break
      print(i)
```

0
1

2
3
4

Yuxarıdakı nümunədə siyahı tərtib etdik.daha sonra for operatoru vasitəsilə siyahı elementlərini i dəyişəninə mənimsətdik.for operatoru dövr operatoru olduğu üçün hər dövrdə i-dəyişəninə siyahı elementlərini tək tək mənimsədir.if şərt operatoru daxilində i-nin dəyəri 5 ilə bərabərlik müqayisəsində proqramı yarıda kəsməyi təmin edirik.proqramı sonlandıran break ifadəsi dövrün 5==5 şərtində proqramı sonlandırır

```
[33]: siyahı = [*range(int(input('Num :')))]  
print(siyahı)
```

Num :11
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
[36]: siyahı = [*range(int(input('Num :')))]  
for i in siyahı:  
    if i == 7:  
        print("Item found!")  
        break  
    print(i)  
else:  
    print("Item not found!")
```

Num :11
0
1
2
3
4
5
6
Item found!

```
[ ]:
```

```
[ ]:
```

while operatoru ilə istifadə üçün nümunələrə nəzər yetirək

```
[24]: while True: # Sonsuz dövr,hər zaman doğrudur  
    password = input("Şifrənizi qeyd edin (proqramdan çıxmaq üçün 'q'-y' basın):  
    ↵")  
    if (password == "q"):  
        print("Proqramdan çıxılır ...")  
        break  
    elif (len(password)<8):
```

```

    print('Şifrəniz 8-saydan az olmamalıdır.')
elif (len(password)>=8):
    print('Şifrəniz qəbul olundu.Təşəkkürlər!')
    break
print('*'*30)

```

Şifrənizi qeyd edin (proqramdan çıxmaq üçün 'q'-y' basın):q
Proqramdan çıxılır ...

[]:

```

[26]: while True: # Sonsuz dövr,hər zaman doğrudur
    password = input("Şifrənizi qeyd edin (proqramdan çıxmaq üçün 'q'-y' basın):
    ↵")
    if (password == "q"):
        print("Proqramdan çıxılır ...")
        break
    elif (len(password)<8):
        print('Şifrəniz 8-saydan az olmamalıdır.')
    elif (len(password)>=8):
        print('Şifrəniz qəbul olundu.Təşəkkürlər!')
        break
print('*'*30)

```

Şifrənizi qeyd edin (proqramdan çıxmaq üçün 'q'-y' basın):qwerty
Şifrəniz 8-saydan az olmamalıdır.
Şifrənizi qeyd edin (proqramdan çıxmaq üçün 'q'-y' basın):qwerty1987
Şifrəniz qəbul olundu.Təşəkkürlər!

```

[53]: i = 0
while i < 10:
    print('i-nin dəyəri:',i)

    if i == 5:
        break

    i += 1

```

i-nin dəyəri: 0
i-nin dəyəri: 1
i-nin dəyəri: 2
i-nin dəyəri: 3
i-nin dəyəri: 4
i-nin dəyəri: 5

Yuxarıdakı nümunədə i-nin dəyərini 0 edib dövrü 10 ədədinə qədər davam etməsini şərt olaraq təmin etdik.Öncə i = 0 olduğundan 10-dan kiçikdirmi şərti ödənilir və ekrana i-nin dəyəri çap olunur.Daha

sonra if şərt operatoru ilə i-nin dəyərini 5-ilə bərabərliyini yoxlayırıq,təbiki şərt ödənilmədiyindən artıq break ifadəsi öz işini görmür və if blokundan çıxılır.if bloku tamamlandıqından son ifadə i-nin 1-vahid artırılması tətbiq olunur və ardından dövr ən başa dönür.artıq i-nin dəyəri 1 olduğunu bilirik.Eyni ardıcılıqla dövr təkrarlanır.O vaxta qədərki i-nin dəyəri 5 olduqda,if blokunda 5==5 şərti ödəndiyindən artıq break ifadəsi öz işini görərək proqramı sonlandırır.

```
[52]: siyahı = []
      for var in "pythonista":
          if var == "i":
              break
          siyahı.append(var)
      string = "".join(siyahı)
      print(var)

      print("Bitdi....{} dili".format(string))
```

```
p
y
t
h
o
n
Bitdi...python dili
```

```
[ ]:
```

2.0.5 continue ifadəsi

continue ifadəsi break ifadəsinə nəzərən az istifadə olunan ifadədir.Çünki continue ifadəsinə əvəzedici digər operatorları istifadə etmək mümkündür.Nümunələrə nəzər yetirək

```
[60]: siyahı = [*range(11)]
      for i in siyahı:
          print('i:',i)
```

```
i: 0
i: 1
i: 2
i: 3
i: 4
i: 5
i: 6
i: 7
i: 8
i: 9
i: 10
```

Yuxarıdakı nümunədə siyahı tərtib edib for operatoru ilə dövr tərtib edərək hər dövrdə i-dəyişəninə siyahının elementlərini mənimsəyib ekrana çap etdik

for bloku daxilində şərt operatoru tərtib edib continue ifadəsindən istifadə etmək

```
[62]: siyahı = [*range(6)]
      for i in siyahı:
          if (i==2):
              continue;
          print('i:',i)
```

```
i: 0
i: 1
i: 3
i: 4
i: 5
```

Yuxarıdakı kod nümunəsində if şərt operatoru ilə i-nin dəyəri dövr təkrarlandıqca 2-yə bərabər olduğu müddətdə continue ifadəsi dövr daxilində öz işini görür. ifadədən sonra (şərt operatoru bloku) digər çalışdırılacaq kod olmadığından blok tamamlanıb başa döndüyündən dolayı print() funksiyası bu keçiddə i-nin dəyəri, ikini ekrana çap etmədi

```
[69]: siyahı = []
      for var in "ələk":
          if var == "k":
              continue
          siyahı.append(var)
      string = "".join(siyahı)
      print(var)
      print('*'*30)
      print("{} ".format(string))
```

```
ə
l
ə
*****
ələ
```

continue ifadəsi sonsuz dövrlərə zəmin yarada bilər. Aşağıdakı nümunədə proqramın çökməsinə səbəb olacaq

```
[70]: counter = 0;
      while (counter<10):
          if (counter == 3):
              continue
          print(counter)
          counter+=1
```

```
0
1
2
```

```

-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-70-b69bc3438d54> in <module>
      1 counter = 0;
----> 2 while (counter<10):
      3     if (counter == 3):
      4         continue
      5     print(counter)

KeyboardInterrupt:

```

```

[72]: counter = 0;
      while (counter<10):
          if (counter == 3):
              counter+=1  #sonsuz dövrü dayandırmaq üçün 3-ədəbindən növbəti ədədə
          ↪ keçid edirik
              continue
          print(counter)
          counter+=1

```

```

0
1
2
4
5
6
7
8
9

```

```
[ ]:
```

2.0.6 pass ifadəsi

ötürmə keçid mənalarmı ifadə edir. Bu ifadənin icrası zamanı heç bir nəticə olmur və bu cəhətdən zəruri olduğu yerlərdə istifadə olunur.

```

[85]: for i in list(range(10)):

      if i == 5:
          pass
          print ('Pass ifadəsi')
      print ('dədlər :', i)

```

```

dədlər : 0
dədlər : 1

```

```
dədlər : 2  
dədlər : 3  
dədlər : 4  
Pass ifadəsi  
dədlər : 5  
dədlər : 6  
dədlər : 7  
dədlər : 8  
dədlər : 9
```

Yuxarıdakı nümunədə 10-a qədər siyahı daxilində ədədlər diapazonu təşkil etdik,şərt operatoru ilə bərabərliyin ödəndiyi təqdirdə pass ifadəsi və icra olunacaq növbəti print() funksiyası icra olunur.continue ifadəsindən fərqli olaraq ifadə kod sətrində özündən sonra gələn kod sətrilərinin çalışmasına maneə törətmir.

[]: