

- ✓ Proqramda verilənləri kəmiyyət adlandırırlar.
- ✓ Proqramda qiymətini dəyişən kəmiyyətlərə **dəyişənlər** deyilir.
- ✓ Dəyişməyənlərə isə **sabitlər və ya konstantlar** deyilir.
- ✓ Hər hansı dəyişən yaratmaq üçün **bərabərlik(=)** işarəsindən istifadə edirik.
- ✓ Məs:z=6.burada o bildirilir ki, z dəyişəni 6 qiymətini alıb.
- ✓ Dəyişənin qiymətini nə olduğunu bilmək istəyirsinizsə komanda sətrində **print** və ondan sonra mötərizə içərsində dəyişənin adını yazıb **Enter** klavişini sıxın.

```
>>> z=6
>>> print(z)
6
>>> |
```

- ✓ Dəyişənin adında latın əlifbasının həm böyük ,həm də kiçik hərflərindən ,rəqəmlərdən və alt cizgidən (_) istifadə etməyə icazə var.**Birinci rəqəm ola bilməz.**Alt cizgi dəyişənin adında əvvəldə,ortada,sonda gələ bilər.

```
>>> a=5
>>> b=1000
>>> Amin="muellim"
>>> 5a=3
SyntaxError: invalid decimal literal
>>> a_b=7
>>> _ab=5
>>> ab_=6
>>> |
```

- ✓ Hərfdən sonra istədiyiniz qədər rəqəm yaza bilərsiniz,amma rəqəmlə başlayan dəyişən adı yaza bilmərsiniz.
- ✓ Alt cizgidən istifadə edə bilərsiniz,amma -(çıxma) işarəsindən istifadə edə bilmərsiniz.

```
>>> a_b=5
>>> a-b=3
SyntaxError: cannot assign to expression here. Maybe you meant '==' instead of '
='?
>>> |
```

- ✓ Şərhlər #işarəsi ilə yazılır.Proqramın hansısa hissəsini müvəqqəti dondurmaq üçün həmin hissə şərh kimi yazılır.

Dəyişənlərə qiymətlər mənimsətməklə biz onlar üzərində riyazi hesablamalar apara bilərik.

```
>>> a=5
>>> b=3
>>> c=a+b
>>> print(c)
8
>>> #yaxud da
>>> a=5
>>> b=3
>>> print(a+b)
8
>>> |
```

- ✓ / -bölmə
- ✓ %-qalıq
- ✓ //-tam hissəni qaytarır

```
>>> a=100
>>> b=3
>>> print(a/b)
33.333333333333336
>>>
>>> print(a%b)
1
>>>
>>> print(a//b)
33
>>> |
```

Müqayisə operatorları

=	Mənimsetmə.yəni harada yazsam ki,a=5,deməli o deməkdir ki,a-ya 5 qiyməti mənimşədilib.
==	Müqayisə deməkdir.Məs:a==b .bu odeməkdir ki,açb-yə bərabərdirmi?
!=	Fərqlidir.yəni a,b-dən fərqlidir.
>	Böyükdür
<	kiçikdir
>=	Böyük bərabərdir
<=	Kiçik bərabərdir

Nümunə:İf şərt operatoru.

#Yadda saxlamaq lazımdır ki,if-lə başlayan sətir mütləq iki nöqtə ilə bitməlidir.

```
a=5
b=3
if a>b:
    print("a b-den boyukdur")
else:
    print("dogru deyil")
```

```
>>> ===== RESTART: C:/Users/User/Desktop/98.py =====
a b-den boyukdur
>>>
```

Və ya bir neçə şərt qoymaq istəsək ,

```
a=8
b=7
if a>b:
    print("a b-den boyukdur")
elif a==b:
    print("a b-ye beraberdır")
else:
    print("a, b-den kicikdir")
```

```
===== RESTART: C:/Users/User/Desktop/98.py =====
a b-den boyukdur
|
```

Data tipləri

Int(integer)

məs:x=5

Str(string)-mətn şəkilli.dırnaqda yazırıq.

məs:x="python"

Float(həqiqi ədəd-kəsr ədəd)

məs:12.3

Biz hər hansı bir datanın tipini müəyyən etmək istəyiriksə bu zaman type() yazmağımız kifayət edər.

```
>>> x=5
>>> type(x)
<class 'int'>
>>> x="python"
>>> type(x)
<class 'str'>
>>> x=12.3
>>> type(x)
<class 'float'>
>>>
>>> |
```

Verilmiş şərtə uyğun ,indexsə görə simvolun müəyyən edilməsi.

```
str1="python"
print(str1[0])
p
print(str1[2])
t
```

Və ya

```
>>> print(str1[0:3])
pyt
>>> |
```

yadda saxla ki,simvollar soldan sağa doğru 0-dan başlayaraq indekslənilir.(nömrələnilir).bu tipli tapşırıqlarda 1-ci mövqenin indeksi nəzərə alınır.amma sonuncu yox.çap zamanı sonuncu götürülmür.Burada qeyd olunub ki,0-cı simvoldan 3-cüyə kimi çap et,amma 3-cünü nəzərə alma.

Və ya

```
>>> print(str1[2:5])
tho
>>> |
```

Və ya

```
a="python"
print(a[2:])
thon
```

bu o deməkdir ki,2-ci simvoldan sonra gələn bütün simvolları çap et.

```
>>> a="program"
>>> print(a[::2])
porm
>>> |
```

bu o deməkdir ki,1-ci simvoldan sonra hər ikincini çapa ver

P r o g r a m

Və ya

```
>>> a="program"
>>> print(a[::-2])
mrop
>>> |
```

burada fikir versəniz addım mənfi 2dir(-2).yəni sağ tərəfdən saymağa başla.yəni,sağ tərəfdən 1-cidən etibarən hər 2-cini çap et.

```
>>> a="program"
>>> print(a[::-2])
mrop
>>> |
```

←
P r o g r a m

Və ya

```
a="program"
print(a[::-3])
mgp
|
```

Və ya müəyyən intervala aid simvolları çap et.

```
>>> a="Azerbaijan"
>>> print(a[2:5:2])
eb
>>> |
```

bu o deməkdir ki,2-ci simvoldan 5-ciyə kimi ,hər 2-cini çap et.

A z e r b a y c a n

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

```
///
>>> a="Qazaxistan"
>>> print(a[-2:6:-2])
a
>>> |
```

- **Upper** – simvolların böyük simvollarla keçməsi

```
>>> a="python"
>>> print(a.upper())
PYTHON
>>> |
```

- **Capitalize-1** –ci simvolun böyük simvola keçməsi

```
>>> a="python"
>>> print(a.capitalize())
Python
>>> |
```

- **Lower-** simvolların kiçik simvola keçməsi

```
>>> A="PYTHON"
>>> print(A.lower())
python
>>> |
```

- **Find-**Verilmiş sözde axtarılan simvolun ilk qarşısına çıxanının indeksini çap edir.

```
>>> a="python"
>>> print(a.find("y"))
1
>>> |
```

P	y	t	h	o	n
0	1	2	3	4	5

```
>>> str1="Azerbaijan"
>>> print(str1.find("a"))
5
>>> |
```

- **Verilmiş indeksə görə siyahıdan elementin çap edilməsi**

```
>>> list1=["sevda", "timur", 20, True]
>>> print(list1[1])
timur
>>> |
```

```
>>> list1=["sevda", "timur", 20, "cebrayil", "elmar", True]
>>> print(list1[1:3])
['timur', 20]
>>> |
```

- **Siyahıya daxil olan elementlərin cəmlənməsi**

SUM-funksiyası

```
>>>
>>> list1=[1,2,3,5,20,-3]
>>> print(sum(list1))
28
```

- **Append funksiyası**-veriləni siyahının sonuna əlavə edir.

```
a=[23,25,"aygun","murad"]
a.append("aynur")
[23,25,,"aygun","murad","aynur"]
```

- **Insert funksiyası** – siyahıda birbaşa təyinatla,hansı mövqeyə deyirsə həmin mövqeyə yerləşdirir.

```
K=["zaur","elmar","sevda"]
k.insert(1,"aqil")
print(k)
["zaur","aqil","elmar","sevda"]
```

- **Count funksiyası** -siyahının elementlərinin içərisindən verilənin sayını tapır

```
>>> a=["a","b","c","a","d"]
>>> print(a.count("a"))
2
```

- **Sort funksiyası** – siyahının elementlərini A-Z düzür.

```
>>> a=["z","e","a","b","c","a","d"]
>>> a.sort()
>>> print(a)
['a', 'a', 'b', 'c', 'd', 'e', 'z']
>>> |
```

!!!

```
>>> a=["z","e","a","b","c","a","d"]
>>> print(a.sort())
None
```

bu onu göstərir ki,sort funksiyası birbaşa çapa verildikdə bizə None qaytarır.

- **Reverse** funksiyası-siyahını tərsinə düzür.

```

list1=["a","b","h","z","l"]
list1.reverse()
print(list1)
['l', 'z', 'h', 'b', 'a']

```

- **Join** funksiyası- siyahının elementlərinin arasına veriləni əlavə edir.

```

list1=["a","b","h","z","l"]
x=";".join(list1)
print(x)
a;b;h;z;l

```

```

>>> k="hello world"
>>> k=k.split()
>>> print("*.join(k))
hello*world
>>>

```

- **Split** funksiyası –siyahının elementlərini ayrı-ayrı çapa verir,deyilən formada həmin yerdən bölür.

```

>>> str1="nuran,ayten, sevda, murad"
>>> a=str1.split(",")
>>> print(a)
['nuran', 'ayten', ' sevda', ' murad']
>>>

```

Tapşırıq1.

```

#siyahıdan "limon"-u çap edin.
list1=[2,["timur","sevda","arzu"],["toyota","lexus"],["portagal","Limon","alma"]]
print(list1[3][1])

```

Tapşırıq 2.

```

# siyahı götürün və onları nizamlayın.alınan neticələri cəmleyin
a=["z","b","c","d"]
b=["m","s","w"]
a.sort()
print(a)
b.sort()
print(b)
print(a+b)

```

Tapşırıq 3

```
#iki list deyisen goturun.I listin ilk 2 elementi ile II listin son 3 elemnetinin cemini tapin.
a=["z","b","c","d"]
b=["m","s","w","f"]
c=(a[:2])
print(c)
d=(b[-3:])
print(d)
print(c+d)
```

Python sadə əməllər

- **Len –funksiyası** simvolları sayır.Bizə ədəd tipli cavab qaytarır.

```
>>> a="informatika"
>>> print(len(a))
11
>>> |
```

- **Sətir tipindən (str) İnt(integer) tipinə dəyişmək** olar.

```
>>> print(int("155"))
...
155
>>> |
```

- **Həqiqi(float) tipdən tam ədəd tipinə keçmək** olar.

```
>>> a=12.5
...
>>> print(int(12.5))
...
12
>>> |
```

- **İnt tipdən həqiq tipə keçmək** olar

```
>>> a=12
>>> print(float(a))
12.0
>>> |
```

- **ədəd şəkilli Str tipdən birbaşa int tipə keçmək mümkün deyil.**Bununçün birinci float(həqiqi) tipə keçib daha sonra int tipə keçmək lazımdır.

```
>>> a="12.5"
...
>>> print(int(a))
...
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    print(int(a))
ValueError: invalid literal for int() with base 10: '12.5'
>>> |
```



```

>>> a="12.5"
>>> b=float(a)
>>> print(b)
12.5
>>> print(int(b))
12
>>> |

```

- Str tipli veriləni hər hansı bir ədədə vurmaq onu o qədər təkrarlamaq deməkdir.

```

>>> a="kitab"
>>> print(a*3)
kitabkitabkitab
>>> |

```

- İki str tipli veriləni toplasaq onlar yanaşı yazılacaq.

```

>>> a="informatika "
>>> b="dersi"
>>> print(a+b)
informatika dersi
>>> |

```

- ədədləri bir-birinə böldükdə bizə həqiqi(float) tip qiymət qaytarır.

```

>>> a=15
>>> b=3
>>> print(a/b)
5.0
>>> |

```

- ədədlər tam bölünmə zamanı int (tam) tipli qiymət alır.

```

>>> a=25
>>> b=3
>>> print(a//b)
8
>>> |

```

bu onu göstərir ki,25-də 3 ,8 dəfədir.

- ədədləri qüvvətə yüksəltmək üçün (iki ulduz-**) və ya pow funksiyasından istifadə edirik.

```

>>> a=5
>>> print(a**3)
125
>>> a=5
>>> b=3
>>> print(pow(a,b))
125
>>> |

```

- Python dilində klaviaturadan a=input() vasitəsilə daxil edilən qiymət a dəyişəninə sətir tipində mənimsədiləcək.Ədəd tipində qəbul etməsi üçün a=int(input())kimi yazılmalıdır.

```

>>> a=input()
a=5
>>> print(type(a))
<class 'str'>
>>> |

```

```
a=int(input())
b=int(input())
print(a+b)
```

```
100
50
150
>>> |
```

- Sep-ayırıcı mənasını verir.Sep-parametrində dırnağın daxilinə nə yazılıbsa siyahı elementlərinə onu tətbiq edəcək,amma ən sonuncu elementə tətbiq etməyəcək.

```
>>> print("informatika","dersi","2023",sep='-')
informatika-dersi-2023
>>>
```

```
>>> print("informatika","dersi","2023",sep='\n')
informatika
dersi
2023
>>>
```

- Sep parametrinin daxilindəki elementlərin arasına əlavə olunurdusa, End –də isə daxilinə nə yazsa,siyahının sonunda o çapa gedəcək gedəcək.

- Print-əmri(nəticənin çap olunması üçün istifadə olunur.

```
>>> a="12"
>>> b="45"
>>> print(a+b)
1245
```

- Sətirlər-dırnağın içərisində istənilən verilən sətirdir.

```
>>>
>>> #Nümunə: "1245","kitab",
```

- Strip-funksiyası boşluqları silir.

```
>>>
>>> a="   informatika   "
>>> print(a.strip())
informatika
>>> |
```

- **Replace-əvəz etmə funksiyası.1-ci veriləni 2-ciyə əvəz edir.**

```
>>> a="abaaaaacddddd"
...
>>> print(a.replace("a","z"))
...
zbzzzzzcddddd
>>> |
```

deməli,əvəz etmə zamanı nələrə əvəz edirsə,onların hamısını atətbəq edir əvəzetməni.

- **Remove-siyahıdan elementi silir.**

```
>>> a=["baki","kitab","muellim"]
>>> b=a.remove("baki")
>>> print(a)
['kitab', 'muellim']
>>>
```

- **Index-siyahıda elementin yerləşmə indeksini çap edir**

```
>>> a=["baki","kitab","muellim"]
>>> print(a.index("kitab"))
1
>>> |
```

Müqayisə operatorları

- **Müqayisə operatorları iki dəyişəni müqayisə edir.Müqayisənin nəticəsini print-də ya True,ya da False cavabını verəcək.**

```
>>> a=5
>>> b=3
>>> a==b
False
>>>
>>> a=5
>>> b=3
>>> c=3
>>> c==b
True
>>>
>>> a!=b
True
>>> a>=b
True
>>> a<=b
False
>>> a>b
True
>>> a<b
False
>>> |
```

- **Abs-modul funksiyası bizə müsbət ədəd verməlidir.**

```
>>> a=-5
>>> print(abs(a))
5
>>>
>>> a=5
>>> print(abs(a))
5
>>> # Deməli, abs funksiyası modul funksiyası olduğundan həm mənfi, həm müsbət ədəddən müsbət ədəd cavabını qaytarır.
```

- **Məntiqi operatorlar(and,or və not operatorları)**
And- və operatoru-hər iki şərt ödənməlidir.

```
///
>>> a=15
>>> b=25
>>> print(a>10 and b>10)
True
>>>
```

Yadda saxlamaq lazımdır ki,bizə qaytarılan cavablar True,True,... kimidirsə,onda cavab bizdə True olacaq.

Yox əgər,True-lərin içində bir dənə False olsa onda cavab bizə False qaydacaq.

Or-və ya.Bir neçə şərtin içində heç olmasa biri ödəyirsə nəticə True verəcək.Yerdə qalan hallarda ,yəni hamısı False olsa cavab False olacaq.

```
>>> a=15
>>> b=25
>>> print(a<b or b>a)
True
>>>
>>> print(a<b or b<a)
True
>>>
>>> print(a<b or a>b)
True
>>>
```

və

```
>>>
>>> print(a<b or b==0)
True
>>> print(a<1 or b<1)
False
>>>
```

Not-inkar operatoru

Yeni nəticə True-dışa False çap olacaq,False-dırsa True.

```
///
>>> a=15
>>> b=20
>>> print(not(a>10 and b>10))
False
>>> |
```

gördüyünüz kimi bütün şərtlər ödənilib ,cavab True-dur.amma inkar operatoruna görə bizə False cavabını qaytarır.

Şert operatoru-budaqlanan alqoritmlərə aiddir.

Dövr alqoritmləri isə-dövrü alqoritmlərə.

İf(əgər)-şert operatoru

Budaqlanmada şert operatorundan istifadə olunur.Şert operatorunun strukturu aşağıdakı kimdir.

If <şert> :

<1-ci fraqment>

Else:

<2-ci fraqment>

!!! yadda saxla ki if-lə başlayan sətir həmişə iki nöqtə ilə bitməlidir.

```
>>> a=15
>>> b=25
>>> if a>b:
...     print("dogrudur")
... else:
...     print("dogru deyil")
... 
```

```
a=int(input())
b=int(input())
if a /b==1:
    print("a b-ye beraberdir")
else:
    print("a b-d'n ferqlidir")
.
```

Budaqlanma zamanı şərtlərdən yalnız biri icra olunur.

Əgərbudaqların sayı çox olarsa,yeni şərtlərimiz çoxalarsa biz if-lərin sayın artırmalıyıq.

Yeni,if-in içinə əlavə if əlavə etməliyik.Biz bunu **elif** kimi yazacağıq.

```
a=int(input())
b=int(input())
if a /b==1:
    print("a b-ye beraberdir")
elif a>b:
    print("a b-den boyukdur")
elif a<b:
    print("a b-d'n kicikdir")
else:
    print("lazim olmayan netice")
```

yadda saxla ki,şert həmişə if-lə başlayır,elif-lə davam edir,else-lə bitir.

