

49–50-DARSLAR. TAKRORLANUVCHI ALGORITMLARNI DASTURLASH. FOR OPERATORI

Berilgan masalani yechishda aynan bir xil amalni bir necha marta takrorlashga toʻgʻri keladi. Bunday masalalarni yechishda takrorlanuvchi algoritmlardan foydalaniladi.

BUNI BILASIZMI?



- 1. Bir necha marta takrorlanadigan kod qanday yoziladi?
- 2. Takrorlanuvchi algoritmlar qanday ishlaydi?
- 3. Sikllar nima uchun kerak?

Takrorlanuvchi algoritmlar – biror buyruqlar guruhining maʼlum marta yoki belgilangan shart bajarilgunga qadar takroran bajarilishi. Takrorlanuvchi algoritmlarga doir masalalarni dasturlashda sikl operatorlaridan foydalaniladi.

Masalan, n ta sonning musbatligini tekshirish uchun n marotaba bir xil amalni bajarish kerak boʻladi. Bunday hollarda bitta amalni n marta yozishdan koʻra, bitta kod blokida n ta sonni tekshirish uchun sikl operatorlaridan foydalangan afzal. Sikl operatorlari kodning takrorlanadigan buyruqlari uchun xizmat qiladi. Bu buyruqlarning ketma-ketligiga *siklning tanasi* deyiladi. Har bir takrorlanish esa *iteratsiya* deb ataladi.

Sikl operatorlari turlari

Sikl operatorlari nomi	Tavsifi	Vazifasi
for	Kodni muayyan marta takrorlaydi.	Takrorlanishlar soni avvaldan maʼlum boʻlganda qoʻllaniladi.
while	Asosiy shart bajarilganda kodni takrorlaydi.	Takrorlanishlar soni nomaʼlum boʻlganda, kod hatto bir marta ham ishga tushmasligi mumkin. Kodni ishga tushirishdan avval shart tekshiriladi. Agar shart notoʻgʻri boʻlsa, unda sikldagi kod ishga tushmaydi.

Qoʻyilgan masalani yechishda sikllarning har ikkala turidan foydalanish mumkin, lekin berilgan shart uchun eng mos keladigan turni toʻgʻri tanlay olish dasturning samaraliroq ishlashini taʼminlaydi.

for sikl operatori

Sintaksisi:
for i in range(start, stop, step):
sikl tanasi

Tayanch tushunchalar

for sikli – hisoblagichli (count-controlled) sikl. U faqat takrorlanishlar soni oldindan maʼlum boʻlganda foydalaniladi.

i – takrorlanishlar (iteratsiyalar) soni;
start – i ning boshlang'ich qiymati (ko'rsatilmasa, 0 deb qabul qiladi);
stop – i ning oxirgi qiymati (ko'rsatilishi shart);
step – qadam (ko'rsatilmasa, 1 deb qabul qiladi);

Misol. 0 dan 11 gacha bo'lgan juft sonlarni chiqarish dasturini tuzing.

```
for i in range(0,11,2):  
    print(i, end=';')  
0;2;4;6;8;10
```

0 dan 11 gacha bo'lgan sonlarni
2 qadam bilan chiqaradi.

Misol. 10 gacha bo'lgan sonlarni chiqarish dasturini tuzing.

```
for i in range(102):  
    print(i, end=';')  
0;1;2;3;4;5;6;7;8;9
```

0 dan 10 gacha bo'lgan sonlarni
1 qadam bilan chiqaradi.

Misol. 10 dan 1 gacha bo'lgan sonlarni –1 qadam bilan chiqarish dasturini tuzing.

```
for i in range(10,0,-1):  
    print(i, end=';')  
10;9;8;7;6;5;4;3;2;1
```

10 dan 0 gacha bo'lgan sonlarni
–1 qadam bilan chiqaradi.

Ichma-ich joylashgan sikllar

Sikl ichida yana siklning ishlatilishiga ichma-ich joylashgan sikl deyiladi.

Sintaksisi:

```
for i in range(start1, stop1, step1):  
    for j in range(start2, stop2, step2):  
        sikl tanasi
```

i – 1-sikl takrorlanishlari soni;
j – 2-sikl takrorlanishlari soni;
start1 – i ning boshlang'ich qiymati (ko'rsatilmasa, 0 deb qabul qiladi);
stop1 – i ning oxirgi qiymati (ko'rsatilishi shart);
step1 – i ning qadami (ko'rsatilmasa, 1 deb qabul qiladi);
start2 – j ning boshlang'ich qiymati (ko'rsatilmasa, 0 deb qabul qiladi);
stop2 – j ning oxirgi qiymati (ko'rsatilishi shart);
step2 – j ning qadami (ko'rsatilmasa, 1 deb qabul qiladi).

Tashqi siklning har bir iteratsiyasi uchun j ta iteratsiya bajariladi. Tashqi siklning i ta iteratsiyasi uchun ichki siklning i*j ta iteratsiyasi bajariladi.

Misol.

```
n = 3
for i in range(1,n+1):
    for j in range(1,n+1):
        print(i,'x', j,'=', i*j)
```

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
```

i va j uchun 3 qadamlar soni kiritiladi.

0 ga ko'paytirishni chiqarmasligi uchun 1 dan boshlanadi, u holda qadamlar soni n+1 deb olinadi. Natija ekranga chiqariladi.

Tashqi siklning 1-takrorlanishida 3 marta ichki sikl bajarilib, natijani chiqardi.

Tashqi siklning 2-takrorlanishi.

Tashqi siklning 3-takrorlanishi.

SAVOL VA TOPSHIRIQLAR



1. Qaysi operator Python dasturlash tilida hisoblagich ko'rinishida ishlaydi?
2. Hisoblagich ko'rinishida ishlovchi sikl operatorining sintaksisi qanday bo'ladi?
3. Start, stop, step vazifalarini tushuntiring.
4. Qachon hisoblagich ko'rinishidagi sikl operatorlaridan foydalanib bo'lmaydi?
5. Ichma-ich joylashgan sikllarda ikkala for operatori bir chiziqda joylashsa, dastur to'g'ri ishlaydimi?

UY VAZIFASI



1. a va b sonlar berilgan. a dan b gacha bo'lgan barcha sonlarni chiqaruvchi dastur tuzing. Bu yerda $a \leq b$.
2. a va b sonlar berilgan. Agar $a < b$ bo'lsa, a dan b gacha bo'lgan sonlarni o'sish tartibida, aks holda kamayish tartibida chiqaring.
3. 1 dan 10 gacha bo'lgan natural sonlar kvadratlarini chiqaruvchi dastur tuzing.
4. 1 dan 10 gacha bo'lgan natural sonlar yig'indisini chiqaruvchi dastur tuzing.