

Python dasturlash tilida for sikl operatori

Rayhon Abdug'afforovna Alimova

Ilmiy rahbar: Chori Begaliyevich Normurodov

Termiz davlat universiteti

Annotatsiya: Ushbu maqolada for sikl operatori, uning imkoniyatlari va python dasturlash tilida qo'llanilishi haqida bayon etildi. Mavzuga oid bir nechta misollarning tahlili qilindi va dasturlari tuzildi.

Kalit so'zlar: For, python, while, sikl, sikl tanasi, break, continue, range(), xrange()

The for loop operator in the Python programming language

Rayhon Abdugafforovna Alimova

Scientific supervisor: Chori Begaliyevich Normurodov

Termiz State University

Abstract: This article describes the for loop operator, its capabilities, and its use in the Python programming language. Several examples of the topic were analyzed and programs were compiled.

Keywords: For, python, while, loop, loop body, break, continue, range(), xrange()

Python dasturlash tilida for operatori C va Paskal dasturlash tillarida qo'llanishidan farq qiladi. Pythonda for operatori biroz murakkabroq, lekin while sikliga qaraganda ancha tezroq bajariladi. For...in operatori obyektlar ketma-ketligida iteratsiyani amalga oshiradi, ya'ni bu sikl har qanday iteratsiya qilinadigan obyekt bo'ylab o'tadi (satr yoki ro'yxat bo'ylab) va har bir o'tish vaqtida sikl tanasini bajaradi. Bir xil hisoblash jarayonlarini bir necha bor takrorlanishi sikl deyiladi. Sikl takrorlanishi davomida bajarilishi lozim bo'lgan operatorlar majmuasi sikl tanasi deyiladi. Sikl tanasi sifatida bir yoki bir nechta operatoridan foydalanish mumkin. Agar sikl tanasida bir nechta operatoridan foydalanmoqchi bo'lsak bu operatorlarni blok {} orasiga olishimiz kerak. Break - funksiyasini har qanday sikl operatoriga qo'llash mumkin. Ya'ni boshqarilishini sikl operatoridan keyingi operatorga uzatadi. Break kalit so'zi bilan siklni to'xtatamiz, hattoki sikl to'xtamagan bo'lsa ham. Masalan, "universitet" so'zining harflarini birma - bir ekranga chiqarish siklini ishga tushuramiz va "s" harfiga yetganda siklni to'xtatamiz:

```

main.py x
1 for x in "universitet":
2     print(x)
3     if x=="s":
4         break
    
```

Endi e'tiborimizni bir narsaga qaratsak. Yuqoridagi kodda print buyrug'i break buyrug'idan oldinroq qo'yg'an edik. Shu sababli avval "s" harfi ekranga chiqib, so'ng sikl to'xtadi. Endi print buyrug'ini pastroqqa qo'yamiz. Bunda "s" harfi ekranga chiqmay qoladi, chunki sikl undan avvalroq to'xtaydi.

```

main.py x
1 for x in "universitet":
2     if x=="s":
3         break
4     print(x)
    
```

Continue - funksiyasini har qanday sikl operatoriga qo'llash mumkin. Bu funksiya sikl parametrining keyingi qiymatini qabul qilishni ta'minlaydi. Boshqacha so'z bilan aytganda sikl tanasi tugatiladi. Bunda esa siklning o'zi tugatilmaydi. Continue kalit so'zi siklning ayrim joylaridan sakrab o'tadi. Aniqroq qilib aytganda sikl davomida ayrim nuqtalarga kelganda ko'rsatilgan amalni bajarmay ketadi. Masalan, "python" so'zidagi harflarni ekranga chiqaramiz va shunda "h" harfini tashlab ketamiz:

```

main.py x
1 for x in "python":
2     if x == 'h':
3         continue
4     print(x)
    
```

Range() funksiyasi biror amalni belgilangan marta bajarish yoki biror oraliqdagi sonlarga murojaat qilish uchun qo'llaniladi. Bunda range() ichiga kerakli son qo'yiladi va sanoq avtomatik tarzda 0 dan boshlanib ko'rsatilgan songacha davom etadi. Ammo uning o'zi hisobga kirmaydi. Masalan, 0 dan 5 gacha (5 soni hisobga kirmaydi) bo'lgan sonlarni ekranga chiqaramiz:

```

main.py x
1 for x in range(5):
2     print(x)
3
    
```

Biz range() funksiyasida sanoq avtomatik 0 dan boshlanishini aytib o'tdik. Biz uni o'zimiz istagan sondan boshlashimiz ham mumkin. Masalan 3 dan 8 gacha bo'lgan sonlarni ekranga chiqaramiz. Bunda sanoq 3 dan boshlanishi uchun 3 sonini ham kiritamiz. Demak, biz 3 dan 9 gacha bo'lgan oraliqni kiritamiz:

```
main.py x
1 for x in range(3, 9):
2     print(x)
3
```

Range() funksiyasida sonlar avtomatik bittaga ortib boradi. Ammo bu holatni ham o'zgartirish mumkin. Bunda oraliqni ko'rsatgandan so'ng sanoq nechtaga ortishini ham kiritamiz. Shunda funksiya ichidagi dastlabki ikkita son oraliqni, uchinchi son esa sanoq nechtaga ortishini ko'rsatadi. Masalan, 1 dan 10 gacha bo'lgan faqat juft sonlarni ekranga chiqarmoqchimiz. Bunda oraliqni 2 dan 11 gacha deb belgilaymiz. Shunda sanoq 2 dan boshlanadi va 10 gacha davom etadi. Har safar sanoq ikkitaga ortishi uchun uchinchi bo'lib 2 soni kiritamiz:

```
main.py x
1 for x in range(2, 11, 2):
2     print(x)
3
```

Katta diapazondagi raqamlardan foydalanilganda range() funksiyasi o'zini oqlamaydi yoki xotira yetishmaydi. Shuning uchun uning o'rniga pythonda xrange() funksiyasidan foydalaniladi.

For operatoriga misollar:

1 - misol. a va b butun sonlari berilgan. (b soni a sonidan katta) a va b sonlari orasidagi barcha butun sonlarni (a va b ni ham) chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi programma tuzilsin.

Dasturi quyidagicha tuziladi:

```
main.py x
1 a=int(input("a butun sonini kiriting: "))
2 b=int(input("b butun (b>a) sonini kiriting: "))
3 for i in range(a, b+1):
4     print(i)
5 print("Chiqarilgan sonlar soni=", (b+1-a))
6
```

Natija oynasi:

```
a butun sonini kiriting: 2
b butun (b>a) sonini kiriting: 6
2
3
4
5
6
Chiqarilgan sonlar soni= 5
```

2 - misol. Bir kilogram konfetning narxi berilgan (haqiqiy son). 1, 3, 5, ... , 15 kg konfetning narxini hisoblovchi dastur tuzilsin.

Dasturi quyidagicha tuziladi:

```
main.py x
1 a=float(input("1 kg konfetning narxini kiriting: "))
2 for x in range(1, 16, 2):
3     s=x*a
4     print(s)
5
```

Natija oynasi:

```
1 kg konfetning narxini kiriting: 2
2.0
6.0
10.0
14.0
18.0
22.0
26.0
30.0
```

3 - misol. n butun soni berilgan (n>0). 1 dan n gacha bo'lgan sonlar ko'paytmasini hisoblovchi programma tuzilsin. 1 dan n gacha bo'lgan sonlar ko'paytmasi n faktorial (n! = 1 * 2 * ... * n) deyiladi.

Quyida dasturi va natija oynasini keltiramiz:

```
main.py x
1 n=int(input("n ni kiriting = "))
2 s=1
3 for i in range(1, n+1):
4     s=s*i
5 print("1*2*...*n=", s)
```

```
n ni kiriting = 4
1*2*...*n= 24
```

Xulosa qiladigan bo'lsam, python dasturlash tilini o'rganish va unda turli masalalarning yechimi topish, dasturlarini tuzish juda ham qulay. Maqolada keltirilgan ma'lumotlar esa pythonni o'rganuvchi va dasturlashga qiziquvchi har qanday shaxs uchun juda muhim hisoblanadi, deb ishonaman.

Foydalanilgan adabiyotlar

1. Sh. A. Mengliyev, O. A. Abdug'aniev, S. Q. Shonazarov, D. Sh. To'rayev: Python dasturlash tili. Termiz-2021.
2. M. R. Fayziyeva, D. M. Sayfurov, N. S. Xaytullayeva. "Informatika va axborot texnologiyalari: umumiy o'rta ta'lim maktablarining 9 – sinfi uchun darslik": Toshkent – 2020.
3. M.T.Azimjanova, M. T. Muradova, M. S. Pazilov. Informatika va axborot texnologiyalari, O'quv qo'llanma, T: O'zbekiston faylasuflari milliy jamiyat nashriyoti – 2013.