

Python programming language

Week 8. Python loops

Alisher Ismailov

Lecturer

Department of Information technology
Andijan Branch of Tashkent Financial Institute

Email: alisherismailov534@gmail.com

Python dasturlash tili

8-Mavzu. Python loops









Ma'ruzachi: Alisher Ismoilov

Axborot texnologiyalari kafedrası
Toshkent moliya instituti Andijon filiali

Elektron pochta: alisherismailov534@gmail.com

8-Mavzu. Python loops

Reja:

-  while sikli
-  For sikli
-  Sikl ichida sikl ishlatish
-  Python iteratorlari (takrorlanuvchilari)
-  Iterator va iterator bo'ladigan obyektlar
-  Loop orqali Iteratordan foydalanish
-  Iterator yaratish
-  Iteratsiyani to'xtatish

Python dasturlash tilida Loops (sikllar)

Python dasturlash tilida
ikkita sikl buyrug'i mavjud:

while sikl

for sikl

Python dasturlash tilida loops (sikllar)

- while sikli bilan biz shart to'g'ri bo'lsa, bir qator holatlarni bajarishimiz mumkin.

```
a = 1
```

```
while a < 5:
```

```
    print(a)
```

```
    a = a + 1
```

Python dasturlash tilida loops (sikllar)

- **break** kalit soʻzi

- **break** kalit soʻzi yordamida **while** sharti toʻgʻri boʻlsa ham siklni toʻxtata olish mumkin:

```
a = 1
```

```
while a < 5:
```

```
    print(a)
```

```
    if a == 2:
```

```
        break
```

```
    a = a + 1
```

Python dasturlash tilida loops (sikllar)

- **continue kalit so'zi**

- continue kalit so'zi bilan sikldagi bitta elementni tashlab keyingi elementdan siklni davom ettirish mumkin:

```
a = 1
```

```
while a < 5:
```

```
    print(a)
```

```
    if a == 2:
```

```
        continue
```

```
    a = a + 1
```

Python dasturlash tilida loops (sikllar)

- **else holati**

- Agar shart to'g'ri bo'lmasa, **else** kalit so'zi bilan biz kod blokini bir marta ishga tushirishimiz mumkin:

```
a = 0
```

```
while a < 5:
```

```
    print(a)
```

```
    a = a + 1
```

```
else:
```

```
    print("a qiymati 5 dan katta")
```

Python dasturlash tilida loops (sikllar)

- **For sikli**

- **For** sikli ketma-ketlikni takrorlash uchun ishlatiladi. **For** sikli yordamida ro‘yxatdagi har bir element uchun bir martadan bajarish mumkin.

```
avtomobil = ["lacetti", "matiz", "spark"]
```

```
for x in avtombil:
```

```
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **String o'zgaruvchini sikl orqali chop etish**
- String o'zgaruvchilar takrorlanadigan obyektlardan biri bo'lib, ular belgilar ketma-ketligini o'z ichiga oladi.

```
kitob = "Python"
```

```
for x in kitob:
```

```
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **break kalit soʻzi**
- Break kalit soʻzi yordamida siklni toʻxtatish mumkin:

```
avtomobil = ["lacetti", "matiz", "spark"]
```

```
for x in avtombil:
```

```
    print(x)
```

```
    if x == "matiz":
```

```
        break
```

Python dasturlash tilida loops (sikllar)

- **continue kalit so‘zi**
- continue kalit so‘zi bilan siklning joriy iteratsiyasini to‘xtatib, keying iteratsiyadan davom ettirish mumkin:

```
avtomobil = ["lacetti", "matiz", "spark"]
```

```
for x in avtombil:
```

```
    if x == "matiz":
```

```
        continue
```

```
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **range() funksiyasi**

- Kodlar to'plamini ma'lum bir necha marta aylantirish uchun **range()** funksiyasidan foydalanish mumkin, **range()** funksiyasi 0 dan boshlanadigan raqamlar ketma-ketligini qaytaradi va 1 ga orttiradi va belgilangan raqamda tugaydi.

```
for x in range(5):  
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **range() funksiyasi**

- **range()** funksiyasi boshlang'ich qiymat sifatida 0 ga teng, lekin parametr qo'shish orqali boshlang'ich qiymatni belgilash mumkin: **range(5, 10)**, bu 5 dan 10 gacha bo'lgan qiymatlarni bildiradi (10 dan tashqari):

```
for x in range(5, 10):  
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **range() funksiyasi**

- **range()** funksiyasi ketma-ketlikni 1 ga oshiradi, biroq uchinchi parametрни qo‘shish orqali o‘shish qiymatini belgilash mumkin: `range(2, 30, 3)`:

```
for x in range(0, 10, 2):  
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **Else holati For sikli Ichida**

- **For** siklidagi **else** kalit soʻzi sikl tugagandan soʻng bajarilishi kerak boʻlgan kod blokini belgilaydi:

```
for x in range(5):
```

```
    print(x)
```

```
else:
```

```
    print("Yakuniga yetdi")
```

Python dasturlash tilida loops (sikllar)

- **Else holati For sikli Ichida**

- Keyingi misolda x o'zgaruvchisi 4ga teng bo'lganda siklni to'xtating va else bloki bilan nima sodir bo'lishini ko'ring:

```
for x in range(5):  
    if x == 4: break  
    print(x)  
else:  
    print("Yakuniga yetdi")
```

Python dasturlash tilida loops (sikllar)

```
haftaKunlari = ["Dushanba", "Seshanba", "Chorshanba",  
"Payshanba", "Juma", "Shanba", "Yakshanba"]  
for x in range(7):  
    print(haftakunlari[x])
```

Python dasturlash tilida loops (sikllar)

```
oynamlari = ["Yanvar", "Fevral", "Mart", "Aprel", "Iyun", "Iyul",  
"Avgust", "Sentabr", "Oktabr", "Noyabr", "Dekabr"]  
for x in range(12):  
    print(oynamlari[x])
```

Python dasturlash tilida loops (sikllar)

```
Dasturlashtillari = ["Java", "C++", "Python", "Javascript"]
```

```
for x in range(4):
```

```
    print(Dasturlashtillari[x])
```

Python dasturlash tilida loops (sikllar)

```
Dasturlashtillari = ["Java", "C++", "Python", "Javascript"]
```

```
for x in range(4):
```

```
    if x == 2: break
```

```
    print(Dasturlashtillari[x])
```

Python dasturlash tilida loops (sikllar)

- **Sikl ichida sikl ishlatish**

- "Ichki sikl" "tashqi sikl" ning har bir iteratsiyasi uchun bir marta bajariladi:

Dasturlashtillari = ["Java", "Python", "C++"]

Sohalar = ["Machine learning", "NLP", "Information retrieval"]

```
for x in dasturlashtillari:
```

```
    for y in sohalar:
```

```
        print(x, y)
```

Python dasturlash tilida Loops (sikllar)

Python iteratorlari (takrorlanuvchilari)

Iterator - bu sanash mumkin bo'lgan qiymatlarni o'z ichiga olgan obyekt hisoblanadi. Iterator takrorlanadigan obyekt, ya'ni siz barcha qiymatlarni birma bir chop etishingiz mumkin.

Python dasturlash tilida iterator `__iter__()` va `__next__()` funksiyalaridan tashkil topgan funksiyalar orqali amalga oshiriladigan obyekt hisoblanadi.

Python dasturlash tilida Loops (sikllar)

Iterator va iterator bo'ladigan obyektlar

Ro'yxatlar, kortejlar, lug'atlar va to'plamlar takrorlanadigan obyekt hisoblanadi. Bu obyektlarning barchasi iteratorni olish uchun ishlatiladigan `iter()` funksiyasiga ega. Misol uchun kortejdan iteratorni funksiyasi orqali har bir qiymatni chop etish jarayoni quyidagicha bo'lishi mumkin:

Python dasturlash tilida loops (sikllar)

- Iterator

```
fanlar = ("fizika", "matematika", "dasturlash")
```

```
iterator = iter(fanlar)
```

```
print(next(iterator))
```

```
print(next(iterator))
```

```
print(next(iterator))
```


Python dasturlash tilida loops (sikllar)

- **Loop orqali iteratordan foydalanish**

Yuqoridagi misollarda iterator qiymatini alohida print() funksiyasi orqali amalga oshirdik. Bu ko'p vaqt va kod yozishni talab qiladigan usul. Bu muammoni loop(sikl) orqali yechish mumkin.

Python dasturlash tilida loops (sikllar)

- **Loop orqali iteratordan foydalanish**

Yuqoridagi misollarda iterator qiymatini alohida print() funksiyasi orqali amalga oshirdik. Bu ko'p vaqt va kod yozishni talab qiladigan usul. Bu muammoni loop(sikl) orqali yechish mumkin.

```
kortej = ("kitob", "daftar", "qalam")
```

```
for x in kortej:
```

```
    print(x)
```

Python dasturlash tilida loops (sikllar)

- **Loop orqali iteratordan foydalanish**

Yuqoridagi misollarda iterator qiymatini alohida print() funksiyasi orqali amalga oshirdik. Bu ko'p vaqt va kod yozishni talab qiladigan usul. Bu muammoni loop(sikl) orqali yechish mumkin.

```
satr = 'kitob'  
for y in satr:  
    print(y)
```

Foydalanilgan adabiyotlar

1. Mastering Object-Oriented Python: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition, Steven F. Lott, Packt Publishing (June 14, 2019)
2. Learning Python, 5th Edition Fifth Edition, Mark Lutz , O'Reilly Media, June 12, 2013
3. Python Programming for Beginners: The Ultimate Guide for Beginners to Learn Python Programming: Crash Course on Python Programming for Beginners, AMZ Publishing, Independently published (July 13, 2021)
4. <https://www.python.org/>
5. <https://www.w3schools.com/>
6. <https://www.codecademy.com/catalog/language/python>
7. <https://realpython.com/>
8. <https://www.anaconda.com/>

**E'tiboringiz
uchun rahmat!**