

PYTHON DASTURLASH TILIDA RANGE() OPERATORI. TO'PLAMLAR

Tojimamatov Isroiljon Nurmamatovich

Farg'ona davlat universiteti

israiltojimatov@gmail.com

Raxmatullayeva Mashxurabonu Qudratillo qizi

Farg'ona davlat universiteti 3-kurs talabasi

raxmatullayevamashxurabonu@gmail.com

Annotatsiya Ushbu maqolada Python dasturlash tilida keng tarqalgan range() operatori haqida batafsil ma'lumot beriladi. Range() funksiyasi, asosan sonlar ketma-ketligini yaratish uchun ishlatiladi va uning imkoniyatlari ko'plab dasturiy algoritmlar, sikllar va iteratsiyalarda muhim ahamiyatga ega. Maqolada range() funksiyasining sintaksisi, ishlatilishi va turli xususiyatlari, shuningdek, uning o'zgarishlar bilan qamday ishlash mumkinligi haqida batafsil tushuntirib berilgan.

Kalit so'zlar: Python, range() operatori, sikl, Iterable, dasturlash, sonlar ketma-ketligi, Iteratsiya.

Annotation This article provides detailed information about the widely used range() operator in Python programming language. The range() function is primarily used to create sequences of numbers, and its capabilities are of significant importance in many programming algorithms, loops, and iterations. The article explains in detail the syntax, usage, and various features of the range() function, as well as how it can work with different variations.

Keywords: Python, range() operator, loop, iterable, programming, sequences of numbers, iteration.

Аннотация В этой статье представлена подробная информация о широко используемом операторе range() в языке программирования Python. Функция range() используется в основном для создания последовательности чисел, и её возможности имеют большое значение во многих программных алгоритмах, циклах и итерациях. В статье подробно объясняются синтаксис, использование и различные особенности функции range(), а также как она может работать с различными изменениями.

Ключевые слова: Python, оператор range(), цикл, итерабельный объект, программирование, последовательности чисел, итерация.

Python dasturlash tili juda qulay va o'qilishi oson sintaksisga ega bo'lib, uning imkoniyatlari katta. Bu tilda turli xil operatorlar va funksiyalar dasturchilarga tez va samarali dasturlar yaratish imkonini beradi. Ularning biri – range() operatoridir. range() funksiyasi Pythonning ichki vositalaridan biri bo'lib, sonlar ketma-ketligini yaratishda va ularni iteratsiya qilishda ishlatiladi. Bu funksiyaning afzalligi shundaki, u xotirani tejashga yordam beradi, chunki bu faqat sonlar ketma-ketligini yaratadi va bu ketma-ketlikni saqlash uchun xotira sarflanmaydi. Ushbu maqolada range() operatorining ishlash prinsiplari, uning sintaksisi va turli parametrlar yordamida qanday ishlatilishini ko'rib chiqamiz.

Ichma-ich joylashgan sikllar.

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.

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

Range() funksiyasining ishlash prinsipi.

Python dasturlash tilida range() funksiyasi ketma-ket sonlar yaratish uchun ishlatiladi.

Uning sintaksisi quyidagicha:

```
python
range(start, stop, step)
```

Bu yerda:

start: Ketma-ketlikning boshlanish qiymati. Agar bu parametr berilmasa, range() funksiyasi 0 dan boshlaydi.

stop: Ketma-ketlik oxiridagi qiymat. Bu qiymat ketma-ketlikka kiritilmaydi.

step: Har bir element orasidagi farq. Bu parametr yordamida ketma-ketlikning qadamini belgilash mumkin.

Quyida range() funksiyasining ishlashiga misollar bilan ko'rib chiqamiz:

1-misol

```
python

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

Bu kod 0 dan 4 gacha bo'lgan sonlarni chiqaradi:

```
0
1
2
3
4
```

2-misol

Bu misolda start va stop qanday ishlashini ko'rib chiqamiz.

```
python

for i in range(2, 6):
    print(i)
```

Natija esa quyidagicha:

```
2
3
4
5
```

3-misol

```
for i in range(1, 10):
    for j in range(1, 10):
        print(i * j, end="\t")
    print("\n")
```

Natija:

THEORY OF SCIENTIFIC RESEARCHES OF WHOLE WORLD

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Step parametri yordamida sonlar ketma-ketligi ortib yoki kamayib borishi mumkin.
Misol uchun, manfiy step yordamida ketma-ketlikni kamaytirish mumkin.

4-misol

Step bilan ishlash.

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

Natija:

```
0
2
4
6
8
```

5-Misol

```
python
for i in range(10, 0, -2):
    print(i)
```

Natija:



10
8
6
4
2

Xulosa: Python dasturlash tilida `range()` funksiyasi butun sonlar ketma-ketligini yaratish uchun ishlatiladi. U odatda `for` tsikllari bilan birga qo'llaniladi va uchta parametrga ega: boshlanish (`start`), tugash (`stop`) va qadam (`step`). Bu funksiya orqali sonlar ro'yxatini yaratish va ularni turli sikllarda avtomatik ishlatish juda qulay.

To'plamlar (`set`) esa Python'dagi tartiblanmagan va takrorlanmaydigan elementlardan iborat ma'lumotlar tuzilmasidir. To'plamlar orqali bir xil qiymatlarni avtomatik ravishda olib tashlash, yangi elementlar qo'shish yoki o'chirish, hamda matematik amallar — birlashtirish, kesish, farq va simmetrik farq bajarish mumkin. To'plamlar sodda va tez ishlovchi tuzilmalar bo'lib, ularni real dasturlarda ma'lumotlar tozaligi va unikalikni ta'minlashda keng qo'llash mumkin.

Umuman olganda, `range()` va `set` Python'da samarali va ixcham kod yozish uchun muhim vositalardan hisoblanadi.

FOYDALANILGAN ADABIYOTLAR

1. "Fluent Python" - Luciano Ramalho
2. "Learning Python" - Mark Lutz
3. Tojimamatov, I., & Mirsiddiqova, M. (2025). Berilganlar bazasida hayotiy sikl. *Модели и методы в современной науке*, 4(6), 66-70.
4. Tojimamatov, I., & Siddiqova, G. (2025). Tranzaksiyalarni taqsimlangan tarzda qayta ishlash modellari. *Современные подходы и новые исследования в современной науке*, 4(6), 30-35.
5. Нурмаматович, Т. И., & Рахила, А. (2025). На основе математических моделей повышение устойчивости к поломкам и авариям. *Yangi o'zbekiston, yangi tadqiqotlar jurnali*, 2(8), 197-204. "Python Crash Course" - Eric Matthes
6. Tojimamatov, I., Ahmataliyeva, S. (2025). Berilganlarni markazlashgan tarzda boshqarish tamoyillari. *Академические исследования в современной науке*, 4(21), 59-64.

7. Tojimamatov, I., & Adxamova, C. (2025). Amaliy tizimlarda berilganlar bazasini boshqarish tizimlari o'rni. Академические исследования в современной науке, 4(21), 77-82.
8. Tojimamatov, I., & Fazliddinov, X. (2025). Berilganlar bazasi administratori va uning xususyatlar. Академические исследования в современной науке, 4(21), 90-95.
9. Karimberdiyevich, O. M., Nurmamatovich, T. I., & Abdulaziz o'g'li, Y. M. (2024). Big data sohasidagi xalqaro loyihalar. Izlanuvchi, 1(1), 39-45.
10. Karimberdiyevich, O. M., Abdulaziz o'g'li, Y. M., & Zarifjon o'g'li, X. N. (2024). Data mining metodlari va bosqichlari. Yangi o'zbekiston, yangi tadqiqotlar jurnali, 1(4), 303-311.
11. Nurmamatovich, T. I. (2024). Berilganlarning tarmoq modellari: oddiy va murakkab tarmoq tuzilishlari.
12. "Python Object-Oriented Programming" - Steven F. Lott
13. "Mastering Object-Oriented Python" - Steven Lott

Foydalanilgan saytlar

1. <https://www.geeksforgeeks.org/method-overriding-in-python/>
2. <https://realpython.com/>
3. <https://www.w3schools.com/python/>
4. <https://docs.python.org/3/tutorial/classes.html>