

PYTHONDA DASTUR YOZISH QOIDALARI

<https://doi.org/10.5281/zenodo.7818599>

Musayev.X.Sh

Toshkent axborot texnologiyalari universiteti

Farg'ona filiali katta o'qituvchi

Abdukarimova.M.S

Andijon viloyati Xo'jaobod tumani 1-umumiy o'rta ta'lim

maktabining 1-toifali informatika fani o'qituvchisi

Annotatsiya. *Ushbu maqolada Pythonda dasturlash tilida dastur yozish qoidalari ko'rib chiqilgan. O'zgaruvchilarni e'lon qilish operatorlarni qanday qo'llash usullari yoritilgan*

Kalit so'zlar. *O'zgaruvchilar, izohlar, shart operatori, Shebang maxsus buyrug'i.*

Аннотация. *В данной статье рассматриваются правила написания программ на языке программирования Python. Объясните, как использовать операторы для объявления переменных*

Ключевые слова. *Переменные, комментарии, условный оператор, специальная команда Shebang.*

Abstract. *This article discusses the rules for writing programs in the Python programming language. Explain how to use operators to declare variables*

Keywords. *Variables, comments, conditional operator, Shebang special command.*

Barcha dasturlash tillarida bo'lgani kabi **Python** dasturlash tilining ham o'ziga mos alifbosi va dastur yozish qoidalari mavjud. **Python** dasturlash tilida har bir buyruqlar yangi satrda joylashtiriladi.

Masalan:

```
print(2 + 3)  
print("Hello")
```

Python dasturlash tilida bo'sh joy va tabulyatsiya juda katta ahamiyatga ega. **Python** dasturlash tilida operatorlar blokini tashkil etish bo'sh joy va tabulyatsiya orqali amalga oshiriladi. Noto'g'ri qo'yilgan bo'sh joy yoki tabulyatsiya xatolik keltiradi. Masalan, yuqoridagi kodni o'zgartirib yoziladigan bo'lsa, xatolik ko'rsatadi.

```
print(2 + 3)  
print("Hello")
```

Shuning uchun yangi ko'rsatmalarni satr boshidan yozish maqsadga muvofiq bo'ladi. Bu hususiyat **Python** dasturlash tilini Java, C# kabi boshqa dasturlash

tillaridan ajratib turadi. Yana bir e'tiborli tomoni shundaki, tilning ba'zi bir operatorlari bir nechta satrlardan iborat bo'lishi mumkin. Masalan:, **if** shart operatori,

```
if a1 < a2:  
print("Hello")
```

Bu holatda `a1` o'zgaruvchisi `a2` o'zgaruvchisidan kichik bo'lsa ekranda **"Hello"** yozuvi hosil bo'ladi. Bu yerda bo'sh joy yoki tabulyatsiya orqali yozilgan `print("Hello")` buyrug'i **if** operatorining bir qismi sifatida kelmoqda. Python dasturlash tilida kodlarni yozishda bo'sh joylarni 4 soniga karrali qilib tashlash maqsadga muvofiq bo'ladi. 5 yoki 6 ta bo'sh joylar tashlansa ham dastur xatosiz ishlaydi. Bunday operatorlar ko'p emas, shuning uchun bo'sh joy tashlash borasida qiyinchiliklar bo'lmaydi. **Python** dasturlash tili C dasturlash oilasidagidek registr tanlaydigan dasturlash tili hisoblanadi, ya'ni bu dasturlash tilida katta va kichik harflar bilan yozilgan o'zgaruvchilar turli xil o'zgaruvchilar deb qaraladi. Shuning uchun dasturlash tilidagi xizmatchi so'zlar ham registrga bog'liq bo'ladi.

Izohlar (kommentariyalar).

Python dasturlash tili yozilgan kodlarga qanday vazifa bajarishini izohlar yordamida yoritib borish mumkin. Dastur ishga tushayotgan vaqtda interpretator izohni taniydi va bu dasturni ishlashiga hech qanday noqulaylik tug'dirmaydi. **Python** dasturlash tili izohlar blokli yoki satrli ko'rinishda bo'ladi. **Python** dasturlash tilida izohlarni (`#` – funta) belgisi bilan hosil qilish mumkin. Blokli izohlar qoldirishda funta belgisini satrning boshiga qo'yiladi.

```
# Habarni konsol oynasiga chiqarish  
print("Hello World")
```

Satrli izohlarda esa buyruqlar yozilgan qator oxiriga yoziladi.

```
print("Hello World") # Habarni konsol oynasiga chiqarish
```

Shebang mahsus buyruq bo'lib, dasturning birinchi satrida izoh sifatida kiritish uchun ishlatiladi. **Shebang** (`#!`) tarzda yoziladi va dastur haqida ma'lumot beradi.

```
#! Pythondagi birinchi dastur  
print("Hello World") #Habarni konsol oynasiga chiqarish
```

Asosiy funksiyalar

Python dasturlash tili bir qatorga kiritiladigan (standart) funksiyalardan iborat. Ulardan ba'zilari dastur yozishda ko'p marta ishlatiladi, ayniqsa ma'lumotlarni ekranga chiqarish va ma'lumotlarni kiritish funksiyalari, lekin ular bilan tanishib chiqamiz.

Ma'lumotlarni chiqarish uchun ishlatiladigan asosiy funksiyalardan biri **"print()"** hisoblanadi:

```
print("Hello Python")
```

Agarda birdaniga bir nechta ma'lumotlarni chiqarish kerak bo'lsa, u holda `print()` funksiyasiga ma'lumotlarni yoki o'zgaruvchilarni, vergul yoki qo'shish (`,` yoki `+`) bilan ajratib yozish mumkin:

```
print("To'liq ismim: " + "Xurshidbek " + "Musayev")
```

Yuqorida keltirilgan dasturning qismi ekranda barcha ma'lumotlarni bir satrda hosil qiladi:

```
To'liq ismim: Xurshidbek Musayev
```

Ma'lumotlarni klaviaturadan kiritish ham mumkin. Bu funksiya `input()` funksiyasi hisoblanadi. Bu funksiyaning ma'lumotlarni kiritish uchun izoh yozish xususiyatiga ega.

```
name = input("Ismingizni kiriting: ")  
print("Salom, " + name)
```

Natija:

```
Ismingizni kiriting: Xurshidbek  
Salom, Xurshidbek
```

O'zgaruvchilar, ma'lumotlar turlari

O'zgaruvchilar ma'lum bir qiymatni o'zida saqlaydi. Python dasturlash tilida boshqa dasturlash tillaridagi kabi lotin harfi yoki ostki chiziqcha bilan boshlanadi. O'zgaruvchilar harflardan yoki raqamlardan tashkil topadi. Bundan tashqari o'zgaruvchilarning nomi Python dasturlash tili xizmatchi so'zlari bilan ustma-ust tushmasligi kerak. Kalit so'zlar dasturlash tilida ko'p emas, ularni eslab qolish oson: **and, as, assert, break, class, continue, def, del, elif, else, expect, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield.**

Masalan: oddiy o'zgaruvchi e'lon qilish quyidagicha amalga oshiriladi:

```
name = "Temur"
```

Bu yerda **Temur** qiymatini o'z ichiga oluvchi **name** nomli o'zgaruvchi aniqlangan. Python dasturlash tilida o'zgaruvchilar ikki xil yo'l bilan nomlanadi: **camel case (tuyasimon)** va **underscore notation (ostki chiziq bilan ajratilgan)**.

Camel case (tuyasimon) da har bir o'zgaruvchini tashkil etuvchi so'zlar katta harflardan boshlanadi. O'zgaruvchining nomi ikki so'zning birikmasidan tashkil topgan bo'lsa, masalan,

```
userName = "Temur"
```

Underscore notation (ostki chiziq bilan ajratilgan) da esa qo'shimcha nomlar ostki chiziq bilan ajratilgan bo'ladi, masalan,

```
user_name = "Temur"
```

Python dasturlash tilida turli xildagi ma'lumotlar turlari mavjud, butun sonli, haqiqiy sonli, kompleks sonlar satrli kattaliklar, ketma-ketliklar, to'plamlar va hokazolarga bo'linadi.

–**boolean** – mantiqiy tip. Qiymati **True** yoki **False**.

–**int** – xotiradan 4 bayt joyini egallaydigan ixtiyoriy butun son.

–**float** – xotiradan 8 bayt joyini egallaydigan ixtiyoriy haqiqiy son.

–**complex** – kompleks son.

–**str** – satrlar. Masalan: "Hello". Python dasturlash tilining 3.x versiyasidan boshlab satrli kattaliklar Unicode belgilarini o'z ichiga olgan.

–**bytes** – nomanfiy butun son. 0 dan 255 gacha bo'lgan sonlarni o'z ichiga oladi.

–**list** – ro'yhat.

–**tuple** – kortej.

–**set** – ob'ektlarning tartibsiz to'plami.

–**frozen_set** – set tipi bilan bir xil. Lekin bu o'zgarmas hisoblanadi.

–**dict** – har bir elementi kalit va qiymatga ega bo'lgan to'plam.

Python dasturlash tilida o'zgaruvchilar ma'lumotlar turini o'zlashtiriladigan qiymatlardan kelib chiqqan holda farqlaydi. Satrlarda qo'shtirnoq (" ") yoki apostrof (' ') lar ichiga yozilgan ma'lumotlar **str** tipiga tegishli deb qaraladi. Agarda o'zgaruvchining qiymati butun son bo'lsa, Python dasturlash tili avtomatik tarzda o'zgaruvchi **int** tipida e'lon qilingan hisoblanadi. Agar o'zgaruvchining qiymati haqiqiy songa tegishli bo'lsa, o'zgaruvchi **float** tipida e'lon qilingan deb qaraladi.

```
x = 3.9e3
print(x)          # 3900.0
x = 3.9e-3
print(x)          # 0.0039
```

Haqiqiy son ko'pi bilan 18 ta raqamdan tashkil topishi mumkin. Agarda raqamlar soni 18 ta raqamdan ko'p bo'lsa, yuqorida ko'rsatilganidek yozish maqsadga muvofiq bo'ladi. Shuning bilan birga bitta dastur yaratilayotgan vaqtda o'zgaruvchilarga bir nechta turdagi qiymatlarni birin-ketin berish mumkin:

```
user_id = "12 Temur smith 438"
print(user_id)
user_id = 234
```

```
print(user_id)
```

`type()` funksiyasi orqali o'zgaruvchilar qaysi tipga tegishli ekanligini bilish mumkin:

```
user_id = "12 Temur smith 438"  
print(user_id)           # <class 'str'>  
user_id = 234  
print(user_id)           # <class 'int'>
```

Arifmetik amallar .Python dasturlash tili barcha arifmetik amallar bilan ishlay oladi.

(+) Ikki sonni qo'shish:

```
print(6 + 2)           # 8
```

(-) Ikki sonni ayirish:

```
print(6 - 2)           # 4
```

(*) Ikki sonni ko'paytirish:

```
print(6 * 2)           #12
```

(/) Ikki sonni bo'lish:

```
print(6 / 2)           # 3
```

(//) Ikki sonni bo'lgandan hosil bo'ladigan sonning butun qismi:

```
print(7 // 2)          # 3
```

(**) Darajaga ko'tarish:

```
print(6 ** 2)          # 36
```

(%) Ikki sonni bo'lgandan hosil bo'lgan qoldiq qismi:

```
print(7 % 2)           # 1
```

Arifmetik amallarni ketma-ket qo'llanilganda ustunlik darajasiga qarab ishlatiladi. Ustunlik darajalari quyidagi ro'yhatda keltirilgan.

1. ** Chapdan o'ngga tomon
2. *, /, //, % Chapdan o'ngga tomon
3. +, - Chapdan o'ngga tomon

Masalan: bizda quyidagi amal bajarilmoqda:

```
number = 3 + 4 * 5 ** 2 + 7  
print(number)          # 110
```

Bu yerda avval darajaga ko'tarish amali bajariladi ($5 ** 2$), so'ngra ko'paytirish amali ($25 * 4$), keyin esa chap tomondagi son qo'shiladi ($100 + 3$) va nihoyat yig'indiga 7 soni qo'shiladi ($103 + 7$) = 110. Amallar ketma-ketligini o'zgartirish uchun qavslardan foydalaniladi:

```
number = (3 + 4) * (5 ** 2 + 7)  
print(number)          # 224
```

Bunday ifodada birinchi o'rinda qavsda ketma-ketliklar bajariladi, avval birinchi qavsning ichi hisoblanadi, so'ng ikkinchi va oxirida ko'paytirish amallari bajariladi. Eslatib o'tish kerakki, arifmetik amallarni bajarayotganda (**int** va **float**) tiplaridan foydalanish mumkin, agarda **int** tipdagi son **float** tipidagi songa qo'shilsa, **int** tipidagi son **float** tipiga o'giriladi.

Qiymat berishli arifmetik amallar

Arifmetik amallarni bajarishda o'zgaruvchining qiymatini qandaydir songa ortirish yoki kamaytirish mumkin. Bu turdagi amallarni qiymat berishli arifmetik amallar deb ataladi. Qiymat berishli arifmetik amallar quyidagilar:

- (+=) Qiymat berishli qo'shish amali;
- (-=) Qiymat berishli ayirish amali;
- (* =) Qiymat berishli ko'paytirish amali;
- (/=) Qiymat berishli bo'lish amali;
- (//=) Qiymat berishli bo'linmani butun qismini topish amali;
- (**=) Qiymat berishli darajaga ko'tarish amali;
- (%=) Qiymat berishli qoldiq topish amali;

```
number = 10  
number += 5  
print(number)          # 15  
number -= 3  
print(number)          # 12  
number //= 5  
print(number)          # 2
```

FOYDALANILGAN ADABIYOTLAR RO'YXATI:

1. R. Zulunov. Preparing the educational process for the era of artificial intelligence. The journal of integrated education and research, Volume 1, issue 4, September 2022, p.261-263.

2. R. Zulunov. Use of artificial intelligence technologies in the educational process. *Web of Scientist: International Scientific Research Journal (WoS)*, Volume 3, Issue 10, Oct., 2022, p. 764-770.
3. Р. Зулунов, Подготовка образовательного процесса к эпохе искусственного интеллекта. *Periodica Journal of Modern Philosophy, Social Sciencies and Humanitiyes*, 2022, Ост., 11, p. 81-83.
4. Р. Зулунов. Что такое искусственный интеллект и как он работает. *Таълим фидойилари*, 2022 ноябр 1 қисм, 149-153 б.
5. R. Zulunov. Sun'iy intellekt texnologiyalarini ta'lim jarayonida qo'llanilishi. *Xorazm Ma'mun akademiyasi habarnomasi*, 11/3 2022, 163-166 b.
6. R. Zulunov, A.Tillavoldiyev. Использование технологий искусственного интеллекта в образовательном процессе. *Periodica Journal of Modern Philosophy, Social Sciences and Humanities*, 2022, v.12, Nov, p.137-142.
7. R. Zulunov, D.Irmatova. Sun'iy intellekt texnologiyalaridan foydalanish. *The journal of integrated education and research*, 1(6), November 2022, p.53-56.
8. R. Zulunov, A. Tadjibaeva. Blended learning in higher education using LMS Moodle. *Obrazovatelnyy protsess*, 2019, 5 (16), p. 5-9.
9. R.Zulunov, S.Xadjyev. Mifologiyadan mashinali o'rganishgacha: sun'iy intellekt tarixi. *Tafakkur manzili*, 2022 dekabr, 1 qism, 16-19 b.
10. R.Zulunov., M.Mahmudova. Sun'iy intellektning insoniyat faoliyatida tutgan o'rni va neyrokibernetika sohasi. *The journal of Integrated Education and Research*, December 2022, 1(7), 2-7 b.
11. Musayev X. S. H., Ermatova Z. Q., Abdurahimova M. I. KOTLIN DASTURLASH TILIDA KLASSLAR VA OB'YEKTLAR TUSHUNCHASI // *Journal of Integrated Education and Research*. – 2022. – T. 1. – №. 6. – С. 126-130.
12. Musayev X. S. H., Ermatova Z. Q. KOTLIN DASTURLASH TILIDA KORUTINLAR BILAN ISHLASHNI TALABALARGA O'RGATISH // *Journal of Integrated Education and Research*. – 2022. – T. 1. – №. 6. – С. 119-125.