ODDIY ARIFMETIK HISOB KITOBLARNI PYTHONDA BAJARISH.

Yusupov Mirsaid Abdulaziz o‘g‘li
Amaliy matematika va informatika
kafedrasi o‘qituvchisi
mirsaidbeky@gmail.com
Xolmatova Dilshodaxon Rasuljon qizi
Farg‘ona davlat universiteti talabasi
xolmatovadilshoda253@gmail.com

Annotatsiya: Ushbu maqola oddiy arifmetik hisob-kitoblarni Python dasturlash tilida qanday amalga oshirishni tizimli tarzda bayon etadi. Maqolada Pythonning asosiy arifmetik operatorlari (+, -, *, /, //, %, **) tushuntiriladi , son turlari (butun va haqiqiy sonlar), operatorlarning ustuvorligi va qavsdan foydalanish, kvadrat ildiz va eksponensial amallar, shuningdek sonlarni formatlash va zaifliklardan (masalan, bo‘lishda ZeroDivisionError) saqlanish usullari ko‘rib chiqiladi. Amaliy qismda oddiy misollar va ularning Python kodlari keltiriladi: summalar, foiz hisoblash, kvadrat va kub hisoblari, butun bo‘linma va qoldiq topish, hamda foydalanuvchidan kiritilgan ma’lumot bilan ishlash misollari. Maqola ta’limiy maqsadga yo‘naltirilgan bo‘lib, boshlovchi dasturchilarga arifmetik mantiqni Pythonda to‘g‘ri va samarali amalga oshirish uchun zarur asosiy prinsiplarni beradi. Xulosa qismida eng ko‘p uchraydigan xatoliklar va ularni oldini olish bo‘yicha tavsiyalar berilgan.

Kalit so‘zlar: Python, arifmetika, operatorlar, butun son, haqiqiy son, operator ustuvorligi, formatlash, istisno (exception), kod misollari, boshlovchilar uchun dars

Abstract: This paper presents a systematic introduction to performing basic arithmetic calculations using the Python programming language. It explains fundamental arithmetic operators (+, -, *, /, //, %, **), numeric types (integers and floats), operator precedence and parentheses, as well as operations like square root and exponentiation. The article covers number formatting and common pitfalls (e.g., ZeroDivisionError) with strategies to handle them. The practical section includes clear code examples for sums, percentage calculations, squares and cubes, integer division and remainder operations, and handling user input. Aimed at beginners, the paper provides essential principles and best practices for implementing arithmetic logic in Python reliably and efficiently. The conclusion summarizes frequent errors and offers advice to avoid them.

Keywords: Python, arithmetic, operators, integer, float, operator precedence, formatting, exceptions, code examples, beginner tutorial

Аннотация: В этой статье систематически изложены основы выполнения простых арифметических вычислений на языке программирования Python.

Рассматриваются базовые арифметические операторы (+, -, *, /, //, %, **), числовые типы (целые и дробные), приоритет операторов и использование скобок, а также операции извлечения квадратного корня и возведения в степень. Показаны приёмы форматирования чисел и обработка распространённых ошибок (например, ZeroDivisionError). В практической части приведены понятные примеры кода для суммирования, расчёта процентов, вычисления квадратов и кубов, целочисленного деления и остатка, а также работы с вводом пользователя. Статья ориентирована на начинающих и содержит рекомендации по предотвращению типичных ошибок.

Ключевые слова: Python, арифметика, операторы, целое число, вещественное число, приоритет операторов, форматирование, исключения, примеры кода, руководство для начинающих .

Zamonaviy dunyoda raqamli texnologiyalar hayotimizning deyarli barcha jabhalariga kirib kelgan. Katta tizimlarni boshqarishdan tortib kundalik oddiy vazifalarigacha texnologiyalar yordamida avtomatlashtirilmoqda. Bunday jarayonning markazida, tabiiyki, dasturlash tillari turadi. Dasturlash tillari orasida o‘zining soddaligi, qulay sintaksisi va keng imkoniyatlari bilan ajralib turadigan Python bugun eng mashhur va ommabop tillardan biriga aylangan. Ayniqsa yangi boshlayotgan dasturchilar uchun Python dasturlashni o‘rganishda qulay boshlang‘ich nuqta bo‘lib xizmat qiladi. Dasturlashni o‘rganish jarayoni odatda eng oddiy elementlardan - matematik hisob-kitoblardan boshlanadi. Chunki arifmetik amallar dastur tuzishning asosiy poydevori sanaladi.

Python tilida oddiy arifmetik amallarni bajarish juda sodda bo‘lib, foydalanuvchi murakkab sintaksis yoki ortiqcha tuzilmalarni yodlab qolishiga hojat yo‘q. Tilning o‘zi matematik amallarni inson uchun qulay shaklda ifodalaydi. Masalan, qo‘shish, ayirish, ko‘paytirish yoki bo‘lish kabi oddiy amallar aynan matematikada qanday yozilsa, Python’da ham deyarli shunday ko‘rinishda yoziladi. Shu sababli ko‘plab o‘quv dasturlar, maktab darslari, kollej va universitetlardagi informatika fanlarida Python tili amaliy mashqlar asosida o‘rgatiladi. O‘quvchi yoki talabalar oddiy arifmetik misollar orqali nafaqat Python sintaksisini, balki dasturlashning asosiy tamoyillarini ham tushunib boradi. Python tilining yana bir afzalligi - uning universalligidir. Dastlab sodda amallarni bajarishni o‘rgangan foydalanuvchi asta-sekin algebraik ifodalar, matematik funksiyalar, statistik tahlillar, grafiklar chizish yoki butun bir ilmiy hisoblash jarayonlarini avtomatlashtirishga o‘tishi mumkin. Pythonning NumPy, Pandas, SymPy kabi kutubxonalari yuqori darajadagi hisoblash imkoniyatini yaratadi, biroq bu bosqichlarga o‘tishdan oldin eng oddiy arifmetik amallarni puxta bilish zarur. Shu sababli Python’dagi oddiy matematik operatorlarni to‘g‘ri tushunish kelajakdagi murakkab dasturlar uchun mustahkam zamin hozirlaydi.

Bugungi kunda deyarli barcha sohalarda - muhandislikdan tortib iqtisodiyotgacha, tibbiyotdan tortib marketinggacha - hisob-kitoblar muhim o‘rin tutadi. Odam tomonidan qo‘lda bajariladigan hisoblar ko‘pincha xato bo‘lishi yoki ko‘p vaqt talab etishi mumkin. Python esa bu jarayonni avtomatlashtiradi, hisob-kitob tezligini oshiradi, aniqlikni esa maksimum darajaga yetkazadi. Masalan, oddiy uy xo‘jaligidagi xarajatlarni hisoblash, o‘quvchilar uchun arifmetik misollar yaratish, kichik tadbirkorlikdagi daromad va zararlarni yuritish, ilmiy eksperiment natijalarini qayta ishlash - bularning barchasida Python orqali oddiy matematik amallar keng qo‘llaniladi. Shuningdek, Pythonning o‘rganishga oson bo‘lishi odamlarni dasturlashga qiziqtiradi. Birinchi darsda darhol murakkab algoritmlar bilan shug‘ullanish o‘rniga, oddiy qo‘shish yoki ayirish amallaridan boshlash o‘quvchini ruhlantiradi, dasturlashga bo‘lgan ishonchini oshiradi. Oddiy matematik amallar orqali dastur kodining qanday ishlashi, kompyuterning buyruqlarni qanday bajarishi, o‘zgaruvchilar va qiymatlar tushunchalari osonroq o‘zlashtiriladi. Shuningdek, arifmetik operatorlar bilan ishlash dasturchining mantiqiy fikrlashi va algoritmik tafakkurini shakllantiradi. Ushbu maqolada Python dasturlash tilida oddiy arifmetik hisob-kitoblarni bajarish jarayoni batafsil yoritiladi. Qo‘shish, ayirish, ko‘paytirish, bo‘lish, qoldiq olish, butun bo‘lish, darajaga oshirish kabi operatorlar va ularning amaliy qo‘llanilishi misollar bilan ko‘rsatiladi. Maqola yangi boshlovchilar uchun mo‘ljallangan bo‘lib, Python tilini o‘rganishning ilk bosqichida yordam beradigan asosiy tushunchalarni tushuntirib beradi. Natijada o‘quvchi dasturlash asoslarini yanada ongli ravishda o‘zlashtirish imkoniga ega bo‘ladi.

Asosiy qism

Python dasturlash tili arifmetik amallarni sodda sintaksis orqali bajarishga mo‘ljallangan bo‘lib, u boshlang‘ich darajadagi o‘rganuvchilar uchun juda qulay. Tilning asosiy operatorlari matematikaning an’anaviy qoidalariga juda yaqin, shu bois Python’da dastlabki amallarni bajarish oson va tabiiy kechadi. Quyida Python’da arifmetik amallar qanday ishlashi, har bir operatorning vazifasi va amaliy misollar orqali batafsil tushuntiriladi.

1. O‘zgaruvchilar (Variables) va ularning roli

Arifmetik hisob-kitoblarni bajarishdan avval, natijalarni saqlash uchun o‘zgaruvchilardan foydalaniladi. O‘zgaruvchi - bu kompyuter xotirasida ma’lumot saqlovchi “quti”. Misol:

a = 12

b = 8

Bu yerda a va b o‘zgaruvchilari mos ravishda 12 va 8 qiymatlarni saqlaydi. Endi ular ustida turli matematik amallar bajarishimiz mumkin.

2. Qo‘shish amali (+)

Qo‘shish – eng ko‘p uchraydigan matematik operatorlardan biri. Pythonda u oddiy matematikadagi kabi + orqali ifodalanadi.

```
a = 7
b = 5
natija = a + b
print("Natija:", natija)
```

Natija: 12

Ushbu operator sonlarni yig‘ish bilan birga, matnlarni birlashtirishda ham ishlatiladi, lekin arifmetik qismda biz uni faqat sonlar uchun qo‘llaymiz.

3. Ayirish amali (-)

Ayirish sonlar orasidagi farqni topish uchun ishlatiladi.

```
x = 20
y = 6
print(x - y)
```

Natija: 14

Agar manfiy natija bo‘lsa ham, Python muammosiz uni chiqaradi.

4. Ko‘paytirish amali (*)

Ko‘paytirish operatori ikki sonning ko‘paytmasini qaytaradi.

```
m = 4
n = 9
print(m * n)
```

Natija: 36

Shuningdek, bu operator matnlarni bir necha marta takrorlash uchun ham ishlatiladi, ammo arifmetik doirada u sonlar ustida qo‘llanadi.

5. O‘nlik bo‘lish (/)

Bo‘lish operatori natijani har doim o‘nlik kasr sifatida chiqaradi, hatto natija butun son bo‘lsa ham:

```
print(15 / 3)  # 5.0
print(17 / 2)  # 8.5
```

Ko‘rinib turibdiki, / operatori float (o‘nlik) qaytaradi.

6. Butun bo‘lish (//)

Agar natijaning faqat butun qismi kerak bo‘lsa, unda // ishlatiladi.

```
print(17 // 2)  # 8
```

Bu operator bo‘linmani pastga qarab yaxlitlaydi.

7. Qoldiq olish (%)

Qoldiq olish operatori bo‘linma qoldig‘ini qaytaradi. Bu, ayniqsa, sonning juft yoki toq ekanligini aniqlash, aylanish algoritmlarida yoki shifrlash jarayonlarida keng qo‘llaniladi.

```
print(17 % 2)  # 1 (toq son)
```

print(24 % 5) # 4

8. Darajaga oshirish (**)

Bu operator sonni istalgan darajaga oshirish imkonini beradi.

print(2 ** 4) # 16

print(5 ** 3) # 125

Darajaga oshirish matematik modellash, fizika va kompyuter grafikasi kabi ko‘plab sohalarda juda muhim.

9. Murakkab arifmetik ifodalar.

Python matematik amallarni tartib bo‘yicha bajaradi. Tartib quyidagicha:

1) **()` ichidagi amallar**
2) **Daraja (**)**
3) **Ko‘paytirish va bo‘lish (*, /, //, %)**
4) **Qo‘shish va ayirish (+, -)**

Misol:

natija = (3 + 2) * 4 - 10 / 5

print(natija)

Bosqichma-bosqich:

(3 + 2) = 5

5 * 4 = 20

10 / 5 = 2

20 - 2 = 18

Natija: 18

10. Foydalanuvchidan qiymat olish va hisoblash.

Arifmetik amallarni real vaziyatlarda qo‘llash uchun foydalanuvchi kiritgan qiymatlar bilan ishlash qulay:

a = int(input("Birinchi sonni kiriting: "))

b = int(input("Ikkinchi sonni kiriting: "))

print("Yig‘indi:", a + b)

print("Ayirma:", a - b)

print("Ko‘paytma:", a * b)

print("Bo‘linma:", a / b)

11. Amaliy qo‘llanish misollari

a) O‘rtacha qiymat hisoblash:

a, b, c = 12, 18, 24

ortacha = (a + b + c) / 3

print("O‘rtacha qiymat:", ortacha)

b) To‘rtburchak yuzini hisoblash:

uzunlik = 7

eni = 5

```
yuzasi = uzunlik * eni
print("Yuza:", yuzasi)
```

c) Valyuta konvertatsiyasi:

```
kurs = 12600  # 1 USD = 12600 so‘m
usd = 50
so‘m = usd * kurs
print("50 USD =", so‘m, "so‘m")
```

Xulosa qilib aytganda, Python dasturlash tilida oddiy arifmetik amallarni bajarish - dasturlashning eng muhim va asosiy bosqichlaridan biridir. Chunki aynan shu sodda amallar orqali foydalanuvchi Python sintaksisini, o‘zgaruvchilar bilan ishlashni, operatorlar tartibini va dastur mantiqini anglay boshlaydi. Arifmetik amallar dasturlashning poydevor bo‘lib, ular asosida murakkab matematik funksiyalar, ilmiy hisob-kitoblar, algoritmlar va hatto sun’iy intellekt modellarigacha bo‘lgan barcha jarayonlar shakllanadi. Python tilining soddaligi va qulayligi arifmetik amallarni yanada osonroq bajarish imkonini beradi. Bu esa yangi o‘rganuvchilarga o‘z bilimlarini mustahkamlashda katta yordam beradi. Operatorlarning matematikaga o‘xshash tarzda yozilishi dasturchiga intuitiv tushunishni ta’minlaydi. Natijada o‘quvchi dasturlashning boshlang‘ich qadamlari - qo‘shish, ayirish, ko‘paytirish, bo‘lish, qoldiq olish, darajaga oshirish kabi asosiy amallarni bemalol bajarib, keyinchalik murakkab mavzularni osonroq o‘zlashtira oladi. Shuningdek, Python orqali bajarilgan oddiy amallar real hayotdagi ko‘plab vazifalarning avtomatlashtirilishiga zamin yaratadi. Kundalik hisob-kitoblar, iqtisodiy tahlillar, texnik loyihalar, ilmiy tajribalar natijalarini qayta ishlash - bularning barchasi aynan oddiy matematik operatorlar asosida shakllanadi. Shu bois mazkur mavzuni chuqur o‘zlashtirish nafaqat nazariy bilim beradi, balki amaliy ko‘nikmalarni ham rivojlantiradi. Umuman olganda, Python’da oddiy arifmetik amallarni bajarishni bilish - dasturlash olamiga kirishning eng yaxshi eshigi. Bu bilim kelajakda murakkab loyihalar, algoritmlar, ma’lumotlar tahlili va sun’iy intellekt kabi sohalarga bemalol o‘tish imkonini beradi. Demak, ushbu mavzu har bir yangi boshlovchi dasturchining mustahkam tayanchiga aylanishi shubhasiz.

Foydalanilgan adabiyotlar:

1. Lutz, M. Learning Python. 5th Edition. O’Reilly Media, 2013.
2. Sweigart, A. Automate the Boring Stuff with Python. No Starch Press, 2019.
3. Zed A. Shaw. Learn Python the Hard Way. Addison-Wesley, 2014.
4. Beazley, D. Python Essential Reference. 4th Edition. Addison-Wesley, 2009.
5. Van Rossum, G., & Drake, F. L. The Python Language Reference Manual. Python Software Foundation, 2015.
6. Python.org - Rasmiy Python hujjatlari. https://docs.python.org

7. Ramadanov, R. va boshqalar. Python dasturlash tili asoslari. Toshkent: Innovatsion rivojlanish nashriyoti, 2021.
8. Sweigart, A. Invent Your Own Computer Games with Python. No Starch Press, 2016.
9. N. N. Mirzayev. Dasturlash asoslari: Python tili misolida. Toshkent: TDIU nashriyoti, 2020.
10. McKinney, W. Python for Data Analysis. O'Reilly Media, 2018.