



PYTHON DASTURIDA SONLAR USTIDA ARIFMETIK AMALLAR

Tojimamatov Israiljon Nurmatovich

Farg'ona davlat universiteti o'qituvchisi

E-mail: israiltojimatov@gmail.com

Abdusamadova Vasila Elyorjon qizi

Farg'ona davlat universiteti talabasi

E-mail: abdusamadovavasila@gmail.com

Annotatsiya: Ushbu maqolada Python dasturlash tilida bajariladigan asosiy arifmetik amallar yoritiladi. Har bir amal — qo'shish, ayirish, ko'paytirish, bo'lish, butun bo'lish, qoldiq olish va darajaga ko'tarish — tushunarli misollar orqali batafsil tushuntiriladi. Shuningdek, *int* va *float* kabi sonli ma'lumot turlarining arifmetik ifodalardagi o'rni hamda nolga bo'lish kabi keng tarqalgan xatoliklar haqida alohida e'tibor qaratiladi.

Maqolaning asosiy maqsadi boshlang'ich dasturchilarga Python tilida matematik amallarni to'g'ri va samarali bajarishni aniq va amaliy tarzda o'rgatishdan iborat. Natijada o'quvchilar sonli ma'lumotlar bilan ishlash va dasturlarda asosiy hisoblash mantiqini amalga oshirish uchun zarur bo'lgan ko'nikmalarga ega bo'ladilar.

Kalit so'zlar. Python dasturlash tili, arifmetik amallar, sonli ma'lumot turlari, qo'shish, ayirish, ko'paytirish, bo'lish, butun bo'lish, qoldiq olish, darajaga ko'tarish, dasturlashda xatoliklar, boshlang'ich dasturchilar

Annotation: This article explores the fundamental arithmetic operations in the Python programming language. Each operation—addition, subtraction, multiplication, division, integer division, modulus, and exponentiation—is explained in detail using illustrative examples. Special attention is given to the role of numeric data types such as *int* and *float* in arithmetic expressions, as well as to common errors like division by zero.



The primary objective of the article is to provide beginner programmers with a clear and practical understanding of how to correctly and efficiently perform mathematical operations in Python. As a result, readers will gain essential skills required for working with numerical data and implementing basic computational logic in their programs.

Keywords: *Python programming language, arithmetic operations, numeric data types, addition, subtraction, multiplication, division, integer division, modulo, exponentiation, programming errors, beginner programmers*

Аннотация: *В данной статье рассматриваются основные арифметические операции, выполняемые с числами в языке программирования Python. Каждая операция — сложение, вычитание, умножение, деление, целочисленное деление, взятие остатка и возведение в степень — подробно объясняется на понятных примерах. Особое внимание уделяется роли числовых типов данных `int` и `float` в арифметических выражениях, а также распространённым ошибкам, таким как деление на ноль.*

Основная цель статьи — предоставить начинающим программистам ясное и практическое понимание правильного и эффективного выполнения математических операций в Python. В результате читатели приобретут необходимые навыки для работы с числовыми данными и реализации базовой вычислительной логики в своих программах.

Ключевые слова: *Язык программирования Python, арифметические операции, числовые типы данных, сложение, вычитание, умножение, деление, целочисленное деление, взятие остатка, возведение в степень, ошибки программирования, начинающие программисты*

Kirish

Zamonaviy dasturlash tillari ichida Python o'zining soddaligi, o'rganish osonligi va keng imkoniyatlari bilan ajralib turadi. Ayniqsa, matematik hisob-kitoblar va arifmetik amallarni bajarishda Python dasturchilar uchun ishonchli vositadir. Python dasturlash tili ko'plab sohalarda, jumladan ta'lim, ilm-fan, iqtisod,



muhandislik va boshqalarda keng qo'llaniladi. Shuning uchun dasturchilar uchun sonlar ustida arifmetik amallarni to'g'ri va samarali bajarishni bilish juda muhimdir.

Ushbu maqolada Python dasturlash tilida sonlar ustida qanday arifmetik amallar bajarilishi, ularning sintaksisi, qo'llanilishi hamda ularni dasturiy kodda qanday ifodalash mumkinligi misollar orqali batafsil ko'rib chiqiladi.

Dolzarblik

Bugungi kunda har qanday dasturlash loyihasida matematik amallar bajarilishi zarurati mavjud. Bu, ayniqsa, moliyaviy tahlillar, ilmiy tadqiqotlar, grafikalar, o'yinlar va boshqa ko'plab sohalarda sezilarli ahamiyatga ega. Python tili dasturlashning boshlang'ich bosqichida o'rganilayotgan tillardan biri bo'lib, u sonlar ustida arifmetik amallarni bajarish imkoniyatlarini keng taqdim etadi. Dasturlashni endigina o'rganayotganlar uchun bu amallarni to'g'ri tushunish va qo'llash dasturlash ko'nikmalarining asosiy qismidir.

Maqsad va vazifalar

Ushbu maqolaning asosiy maqsadi Python dasturlash tilida sonlar ustida bajariladigan arifmetik amallarni keng va aniq tushuntirishdan iborat. Shu maqsadda quyidagi vazifalar bajarildi:

- Python tilida mavjud arifmetik amallar turlarini aniqlash;
- Har bir amalning sintaksisi va ishlash tartibini ko'rsatish;
- Amaliy misollar orqali arifmetik amallarni tushuntirish;
- Ma'lumot turlari (int va float) hamda ularning amallarga ta'sirini yoritish;
- Bo'lish operatsiyasidagi xatoliklar va ulardan qochish yo'llarini tushuntirish.

Asosiy tushunchalar

Python dasturlash tilida sonlar ustida quyidagi arifmetik amallar keng qo'llaniladi:

Qo'shish	+	Ikkita sonni qo'shadi	$5 + 3 \rightarrow 8$
Ayirish	-	Bir sonni ikkinchisidan ayiradi	$10 - 4 \rightarrow 6$



Ko'paytirish	*	Ikkita sonni ko'paytiradi	$6 * 7 \rightarrow 42$
Bo'lish	/	Haqiqiy bo'linmani hisoblaydi	$8 / 2 \rightarrow 4.0$
Butun bo'lish	//	Bo'linmaning butun qismini oladi	$9 // 4 \rightarrow 2$
Qoldikli bo'lish	%	Bo'linmadan qolgan qoldiqni beradi	$10 \% 3 \rightarrow 1$
Darajaga ko'tarish	**	Sonni darajaga ko'taradi	$2 ** 3 \rightarrow 8$

Bu amallar yordamida Python dasturchilari har qanday matematik hisob-kitoblarni bajarishi mumkin.

Misollar orqali tushuntirish.

Quyida Python kodlari yordamida har bir arifmetik amal ko'rsatib o'tilgan:

Asosiy arifmetik amallar

$a = 15$

$b = 4$

`print("Qo'shish:", a + b)` # Natija: 19

`print("Ayirish:", a - b)` # Natija: 11

`print("Ko'paytirish:", a * b)` # Natija: 60

`print("Haqiqiy bo'lish:", a / b)` # Natija: 3.75

`print("Butun bo'lish:", a // b)` # Natija: 3

`print("Qoldiq:", a % b)` # Natija: 3

`print("Darajaga ko'tarish:", a ** b)` # Natija: 50625

Ushbu misollarda ko'rilayotgan amallar Python dasturida qanday bajarilishi aniq ko'rsatilgan.

Arifmetik operatorlarning ma'lumot turlariga ta'siri. Python'da arifmetik operatorlar operandlarning ma'lumot turiga qarab o'zgarishi mumkin. Quyidagi holatlar alohida e'tiborga loyiq:



- ✚ $\text{int} + \text{float} \rightarrow \text{float}$: ya'ni, aralash turlarda float ustunlik qiladi.
- ✚ $\text{str} + \text{str} \rightarrow$ satrlarni birlashtiradi, ammo $\text{str} + \text{int}$ xatolik beradi.
- ✚ $\text{list} * \text{int} \rightarrow$ ro'yxatni ko'paytiradi, lekin $\text{list} * \text{list}$ noto'g'ri hisoblanadi.

Raqamlar bilan ishlashda operator ustuvorligi (priority). Python'da ifodalarni yozishda arifmetik amallar qanday ketma-ketlikda bajarilishi muhim. Quyidagi jadval operatorlar ustuvorligi (prioriteti)ni ko'rsatadi:

Ustuvorlik darajasi	Amal	Operator
1 (eng yuqori)	Darajaga ko'tarish	**
2	Ko'paytirish, bo'lish, butun bo'lish, qoldiq	*, /, //, %
3	Qo'shish va ayirish	+, -

Misol: $\text{natija} = 2 + 3 * 4 ** 2 \# 2 + 3 * 16 = 2 + 48 = 50$

Agar boshqa ketma-ketlikda hisoblashni istasak, qavsdan foydalanamiz:

$\text{natija} = ((2 + 3) * 4) ** 2 \# (5 * 4)^2 = 20^2 = 400$

Noma'lumotli (undefined) holatlar va istisnalar. Arifmetik amallarni bajarishda ayrim holatlar istisnoga olib keladi. Masalan:

- Nolga bo'lish (/ yoki //): Python ZeroDivisionError xatosini chiqaradi.
- Juda katta darajalar ($a ** b$) natijasi OverflowErrorga olib kelishi mumkin.

try:

`a = 10`

`b = 0`

`print(a / b)`

except ZeroDivisionError:

`print("Xatolik: nolga bo'lish mumkin emas.")`

O'zgaruvchilar va ma'lumot turlari.

Python dasturlash tilida sonlar int va float turlarida bo'ladi. int — butun sonlarni, float esa haqiqiy sonlarni ifodalaydi. Arifmetik amallarni bajarishda bu turlar o'zgarishi mumkin:



```
x = 5    # int
```

```
y = 2    # int
```

```
z = x / y # float, chunki haqiqiy bo'lish amalida natija float bo'ladi
```

```
print(type(z)) # <class 'float'>
```

Shuni ta'kidlash kerakki, bo'lish amali (/) har doim float natija beradi, hattoki ikkala operand ham butun son bo'lsa ham.

Xatoliklardan ehtiyot bo'lish.

Python dasturlash tilida ba'zi arifmetik amallar noto'g'ri bajarilganda xatoliklar yuzaga keladi. Masalan, nolga bo'lish taqiqlangan:

```
print(5 / 0) # ZeroDivisionError: division by zero
```

Bu holatda dastur "ZeroDivisionError" xatosini beradi. Dasturchilar bunday holatlardan qochish uchun nolga bo'lishni oldindan tekshirishlari lozim:

```
a = 5
```

```
b = 0
```

```
if b != 0:
```

```
    print(a / b)
```

```
else:
```

```
    print("Xato: nolga bo'lish mumkin emas.")
```

Xulosa

Python dasturlash tilida sonlar ustida arifmetik amallarni bajarish juda sodda va tushunarli. Ushbu maqolada keltirilgan misollar orqali boshlovchi dasturchilar Python dasturida asosiy matematik hisob-kitoblarni to'g'ri amalga oshirishga o'rgatiladi. Har bir amal sintaksisi va maqsadi aniq tushuntirilgan, shuningdek, dasturda yuzaga kelishi mumkin bo'lgan asosiy xatoliklar haqida ogohlantirish berilgan. Bu bilimlar dasturlash ko'nikmalarini mustahkamlashda muhim ahamiyatga ega.

Kelajakda Python tilining matematik imkoniyatlari va kutubxonalar (masalan, NumPy, SciPy) orqali yanada murakkab hisob-kitoblarni bajarish haqida alohida maqolalar yozish mumkin.

**FOYDALANILGAN ADABIYOTLAR**

1. Van Rossum, G., & Drake, F. L. (2009). The Python Language Reference Manual. Network Theory Ltd.
2. Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.
3. Sweigart, A. (2015). Automate the Boring Stuff with Python: Practical Programming for Total Beginners. No Starch Press.
4. Beazley, D. M. (2009). Python Essential Reference (4th ed.). Addison-Wesley Professional.
5. Python Software Foundation. (2024). Python 3.12 Documentation. Retrieved from <https://docs.python.org/3/>
6. Matthes, E. (2022). Python Crash Course (3rd ed.). No Starch Press.
7. Dawson, M. (2010). Python Programming for the Absolute Beginner (3rd ed.). Course Technology PTR.
8. Severance, C. (2016). Python for Everybody: Exploring Data in Python 3. CreateSpace Independent Publishing Platform.