ISSN 3060-4567

RETURN OPERATORI VA FUNKSIYADAN NATIJANI QAYTARISH. FUNKSIYANI BOSHQA FUNKSIYANING TURI, PARAMETRI VA NATIJASI SIFATIDA ISHLATILISHI

Isroiljon Tojimamatov

Farg'ona davlat unversiteti amaliy matematika va informatika kafedrasi katta o'qituvchisi

isik80@mail.ru

No'monova Qoriyaxon Mo'sinjon qizi

Farg'ona davlat unversiteti 3-kurs talabasi

qoriyaxonnomonova@gmail.com

Annotatsiya: *Ushbu maqolada dasturlashdagi asosiy tushunchalardan biri bo'lgan return operatori haqida to'liq va tizimli ma'lumot berilgan. return operatorining funksiyadagi roli, ishlash tartibi, sintaksisi, bir nechta qiymatlarni qaytarish, funksiyalarni parametr yoki natija sifatida ishlatishdagi ahamiyati batafsil yoritilgan. Shuningdek, Python dasturlash tili asosida amaliy misollar orqali tushunchalar mustahkamlangan. Maqola dasturlashni o'rganayotgan talaba va dasturchilar uchun foydali nazariy va amaliy qo'llanma bo'lib xizmat qiladi.*

Kalit so'zlar: *return operatori, funksiyalar, qiymat qaytarish, parametr sifatida funksiya, natija sifatida funksiya, Python dasturlash tili, funktsional dasturlash, tuple qaytarish, closure, shartli return.*

Аннотация*: В данной статье подробно и системно рассматривается одна из ключевых концепций программирования — оператор return. Раскрыты его роль в функциях, порядок работы, синтаксис, возможность возврата нескольких значений, а также использование функций в качестве параметров или результатов других функций. Основные понятия подкреплены практическими примерами на языке программирования Python. Статья представляет собой полезное теоретико-практическое пособие для студентов и программистов, изучающих основы программирования.*

Ключевые слова: *оператор return, функции, возврат значения, функция как параметр, функция как результат, язык программирования Python, функциональное программирование, возврат кортежей, замыкание (closure), условный return*

Annotation: *This article provides a comprehensive and structured explanation of the return operator, one of the fundamental concepts in programming. It explores its role within functions, syntax, operational flow, returning multiple values, and how functions can be used as parameters or returned from other functions. Key concepts are reinforced with practical examples in the Python programming language. The article serves as a valuable theoretical and practical guide for students and developers learning the fundamentals of programming.*

Keywords*: return operator, functions, value return, function as parameter, function as result, Python programming language, functional programming, tuple return, closure, conditional return*

Python dasturlash tilida funksiyalardan foydalanish uchun, albatta, funksiyalarni dastur tarkibida alohida yozilish kerak. Funksiyalar python tilining ixtiyoriy joyida yozilishi mumkin. Funksiyalar dasturchi ishini juda yengillashtiradi. Funksiyalar yordamida programma modullashadi, qismlarga bo‘linadi. Bu esa keyinchalik dasturni rivojlantirishni osonlashtiradi. Dastur yozilish davrida hatolarni topishni yengillashtiradi. Funksiya tarkibiga kiritilishi kerak bo‘lagan o‘zgaruvchi argumentlar, albatta, qavs ichida vergul bilan ajratilib yozilishi kerak.

Funksiya tanasini tasvirlash jarayoni ikki qismdan iborat bo‘ladi, ya’ni funksiya sarlavhasi va funksiya tanasidan iborat bo‘ladi. Funksiya tanasini tasvirlash jarayonida funksiya sarlavhasidan keyin nuqtali vergul qo‘yilmaydi va *def* xizmatchi so‘z orqali boshlanib, natija **return** xizmatchi so‘zidan keyin probel bilan yoziladi.

Funksiya tanasini dastur tarkibiga yozishda funksiyaga murojaat qilishdan bir qadam oldin ixtiyoriy joyda yozish mumkin, murojaatdan keyin yozilsa dastur xatolik qaytaradi. Funksiya tanasi tarkibi yozilishida xuddi boshqa dastur tuzilishi kabi unda ishlatiladigan o‘zgaruvchilar e’lon qilinadi, buyruqlar nuqtali vergul yordamida

ISSN 3060-4567

ajratilish shart emas. Funksiyalarni python dasturlash tilida yozilish jarayonining umumiy ko‘rinishi quyidagicha bo‘ladi:

def <funksiya nomi>(<argumentlar>):

funksiya tanasi

return natija

Misol. *Python dasturlash tilida ikki sonning yig‘indisini hisoblash uchun yig() fuksiya yarating va unga murojat qilishni tasvirlang.*

```
 def       yig(a,b):
y=a+breturn              y
a=input('a=')
b=input('b=')     a=int(a)
b=int(b) z=yig(a,b)
        87
         print(z)
         ===========
RESTART:
C:\Users\User\Desktop\
1.py =============
         a=10
          b=2
          12
         >>>
```

return operatori — bu funksiya bajarilgandan so‘ng natijani tashqariga chiqarish, ya’ni funksiyani chaqirgan joyga qiymat yuborish uchun ishlatiladigan maxsus buyruqdir.

Agar funksiya ichida return operatori mavjud bo‘lsa, u bajariladi va natija qaytariladi. Agar u bo‘lmasa, funksiya avtomatik tarzda Noll qiymatini qaytaradi

return operatorining sintaksisi

```
def funksiya_nomi(parametrlar):
# amallar
```

ISSN 3060-4567

return natija

return ning o‘zidan keyin istalgan turdagi **qiymat, ifoda, o‘zgaruvchi yoki obyekt** bo‘lishi mumkin.

return operatorining asosiy vazifalari

Natijani tashqi muhitga uzatish

Funksiya bajarilgach, return orqali biror qiymat qaytariladi va u chaqiruvchi tomonidan ishlatilishi mumkin.

```
def kvadrat(x):
return x * x
a = kvadrat(5)
 # a ga 25 qiymati yoziladi
```

Funksiyani to‘xtatish

return funksiyani bajarishni shu joyda to‘xtatadi. Undan keyingi kodlar ishlamaydi.

```
def test(x):
if x > 0:
 return "Musbat"
 print("  ")
```

Shartli qaytarish (conditional return)

```
def bahola(ball):
 if ball >= 90:
 return "A" elif ball >= 80:
return "B"
else:
 return "C"
```

Bir nechta qiymatni qaytarish

Python tilida return orqali bir nechta qiymat ham qaytarish mumkin. Ular avtomatik tarzda tuple (to‘plam) sifatida qaytariladi.

```
def hisoblansin(a, b):
 return a + b, a * b
```

yigindi, ko‘paytma = hisoblansin(3, 4)

print(yigindi) # 7

print(ko‘paytma) # 12

return operatorining foydaliligi

-Kengaytirilgan matematik amallarni modullashtirishda

-Ma'lumotlarni qayta ishlash va filtrlashda

-Ko‘p bosqichli algoritmlarda oraliq natijalarni olishda

-Kodni testlash va debug qilishda

Funksiyani boshqa funksiyaning parametri sifatida ishlatish. Python va boshqa ko‘pgina zamonaviy dasturlash tillarida funksiyalar birinchi darajali ob’ektlar hisoblanadi. Bu ular quyidagi xususiyatlarga ega deganidir:

O‘zgaruvchiga tayinlanishi mumkin.

Parametr sifatida uzatilishi mumkin.

Qaytarilishi mumkin.

Funksiyani parametr sifatida boshqa funksiyaga uzatish bu funktsional dasturlashning muhim jihatlaridan biridir. Bu yondashuv amallarni umumlashtirish, kodni modullashtirish va dinamik tarzda funksiyalarni boshqarish imkonini beradi.

Misol

def bajarish(amal, qiymat):

return amal(qiymat) def kvadrat(x):

return x * x print(bajarish(kvadrat, 4))

Natija: 16

Bu yerda kvadrat funksiyasi bajarish funksiyasiga parametr sifatida uzatilgan. bajarish funksiyasi esa uzatilgan funksiyani (ya’ni amal) chaqiradi va unga qiymat ni beradi.

Funksiyani natija sifatida qaytarish

Funksiya boshqa funksiyani natija sifatida qaytarishi ham mumkin. Bunday funksiya ichida yaratilgan boshqa funksiyalar tashqi funksiyadan foydalanadi va ular closure deb ataladi. Bu yondashuv yordamida dinamik tarzda yangi funksiyalar yaratiladi.

Misol

def ko‘paytiruvchi_yarat(n):

def ko‘paytir(x):

return x * n return ko‘paytir ikki_hisobla = ko‘paytiruvchi_yarat(2) print(ikki_hisobla(10))

Natija: 20

Bu yerda ko‘paytiruvchi_yarat funksiyasi ichki ko‘paytir funksiyasini yaratadi va uni natija sifatida qaytaradi. ikki_hisobla orqali 2 ga ko‘paytiruvchi funksiya hosil qilinadi.

Funksiya dasturlashning eng muhim abstraksiyalaridan biridir. return operatori orqali funksiyadan natija qaytarish funksiyaning asosiy vazifalaridan biridir.

Funksiyalarni parametr sifatida uzatish va natija sifatida qaytarish dasturlashda kuchli vositalardan hisoblanadi. Bu yondashuvlar yordamida dastur tuzilmasi ancha ixcham, soddalashgan va qayta ishlatilishga mos bo‘ladi. Python dasturlash tilida bu imkoniyatlar keng qo‘llanilib, funktsional dasturlash paradigmasi bilan mos keluvchi murakkab amallarni soddalashtirish imkonini beradi.

FOYDALANILGAN ADABIYOTLAR:

1. Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.

2. → Python dasturlash tili asoslarini chuqur yorituvchi fundamental qo‘llanma.

3. Sweigart, A. (2019). Automate the Boring Stuff with Python (2nd ed.). No Starch Press.

4. → Amaliy Python dasturlash, funksiyalar va return operatoridan foydalanish.

5. Hetland, M. L. (2017). Beginning Python: From Novice to Professional (3rd ed.). Apress.

6. → Yangi boshlovchilar uchun Python asoslari va funksiyaviy yondashuvlar.

7. Beazley, D. (2009). Python Essential Reference (4th ed.). Addison-Wesley.

8. → Professional darajadagi texnik qo‘llanma, return, closures, va boshqa murakkab mavzular.

9. Python Software Foundation. (2024). The Python Language Reference, version 3.12.

ISSN 3060-4567

10. Retrieved from: https://docs.python.org/3/reference/

11. → Python rasmiy hujjati: return operatorining aniq sintaksisi va ishlash tartibi.

12. Gries, P., Campbell, J., & Montojo, J. (2013). Practical Programming: An Introduction to Computer Science Using Python 3. Pragmatic Bookshelf.

13. → Funksiyalar va ularning ishlash mexanizmlari haqidagi tushunchalar.

14. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.

15. → Funksiya, algoritm tuzilmasi va ularning nazariy asoslari.