

PYTHON DASTURLASHDA MAHALLIY FUNKSIYALAR

Tojimamatov Israiljon Nurmamatovich

Farg'ona davlat universiteti katta o'qituvchisi

israiltojimatov@gmail.com

Ergasheva Mahliyoxon Nuriddin qizi

Farg'ona davlat universiteti 3 – kurs talabasi

mahilyoyoqubova27@gmail.com

Annotatsiya: Ushbu maqola python dasturlash tilida funksiyalar va protseduralar tushunchalarini, ularning yaratilishi, ishlatilishi, parametrlar va argumenlar bilan ishlashni, shuningdek, lokal va global o'zgaruvchilarni boshqarish usullarini batafsil tushuntiradi. Maqolada funksiyalarning dasturdagi ro'lini, takrorlanadigan kod bloklarini qisqartirishdagi ahamiyatini, rekursiya va uning xavflarini, shuningdek, standart funksiyalar misollarini keltiradi.

Kalit so'zlar: Python, funksiya, protsedura, parametr, argument, lokal o'zgaruvchi, global o'zgaruvchi, def, rekursiya, standart funksiyalar.

Annotation: This article explains in detail the concepts of functions and procedures in the Python programming language, their creation, use, working with parameters and arguments, as well as methods for managing local and global variables. The article discusses the role of functions in a program, their importance in reducing repetitive code blocks, recursion and its dangers, as well as examples of standard functions.

Key words: Python, function, procedure, parameter, argument, local variable, global variable, def, recursion, standard functions.

Аннотация: В статье подробно объясняются концепции функций и процедур в языке программирования Python, их создание, использование, работа с параметрами и аргументами, а также методы управления локальными и глобальными переменными. В статье обсуждается роль функций в программе, их важность в сокращении повторяющихся блоков

кода, рекурсия и ее опасности, а также примеры стандартных функций.

Ключевые слова: Python, функция, процедура, параметр, аргумент, локальная переменная, глобальная переменная, def, рекурсия, стандартные функции.

Funksiyalar deyarli har bir python dasturining qurilish bloklari hisoblanadi. Ular haqiqiy harakat sodir bo'ladigan joyda!

print(), len() va round() bu funksiyalarning barchasi o'rnatilgan funktsiyalardir, chunki ular Python tilining o'ziga kiritilgan. Bundan tashqari, muayyan vazifalarni bajaradigan foydalanuvchiga tegishli funktsiyalarni yaratishingiz mumkin

Funksiyalar kodni kichikroq bo'laklarga ajratadi va dastur qayta-qayta ishlatiladigan funktsiyalar uchun juda yaxshi. Dastur har safar bir xil kodni yozish o'rniga, shunchaki funktsiyani chaqiring!

Funksiyalar parametrlar, ya'ni funktsiyaga berilishi mumkin bo'lgan qiymatlar qabul qila oladi va ular ustuda biror amal bajarishi mumkin. Bu parametrlar o'zgaruvchilarga o'xshaydi. Faqat ulardan farqi bu o'zgaruvchilarning qiymati funktsiyani chaqirish vaqtida o'rnatiladi. Funktsiya ish boshlagan vaqtda bularga qiymat biriktirilgan bo'ladi.

Parametrlar funktsiya aniqlanayotgan vaqtda qavs ishida vergul bilan ajratilgan holda ko'rsatiladi. Ularga qiymatni funktsiyani chaqirganimizda biriktiramiz. Ushbu atamalarga e'tibor bering: funktsiya e'lon qilinayotgan vaqtda ko'rsatilgan nomlar parametrlar, funktsiyani chaqirayotganimizda unga berilgan qiymatlar esa argumentlar deyiladi.

Funktsiya – bu ko'p marta ishlatiladigan dastur bo'lagi. Funktsiyalar ma'lum buyruqlar blokini ko'rsatilgan nom bilan saqlash va shu blokni dasturning istalgan joyida, istalgan miqdorda bajarish imkonini beradi.

Dastur tuzish jarayonida ma'lum bir amallar majmuini dasturning turli qismlarida takrorlashga to'g'ri keladi. Dasturning mana shu amallar majmuini o'z ichiga olgan qismi qism dastur deb ataladi. Qism dasturlar ma'lum bir vazifani bajaradi, lekin alohida tizimni tashkil etmaydi.

Qism dasturga murojaat etilganda, unga murojaat etgan asosiy dastur to'xtaydi va boshqaruv qism dasturga o'tadi. Qism dastur bajarilishi tugaganidan so'ng, boshqaruv yana asosiy dasturga qaytadi.

Asosiy dasturda qism dasturlarni chaqirish quyidagi imkoniyatlarni beradi:

- Qism dastur zarurat tug'ilganda chaqiriladi. U ayni bir kodni bir necha marta yozish zaruratini bartaraf qilib, butun dastur davomida ko'p marta foydalanilishi mumkin. Bu kodning blokliligini oshiradi, tushunishni osonlashtiradi va xatolarni topishda yordam beradi.
- Xato bor yoki yo'qligini bitta kod blokining o'zida tekshirsa bo'ladi. Agar xato qism dasturda bo'lsa, faqat qism dasturning o'zini tuzatishga zarurat tug'iladi. Agar qism dasturdan foydalanmasdan, kod bir necha joyda takror-takror yozilsa, u holda butun dastur bo'ylab xatolarni qidirishga to'g'ri keladi.
- Kodni faqat bitta joyda yangilash kerak bo'ladi: Kiritilgan barcha tuzatishlar qism dastur chaqirilishi bilan amal qila boshlaydi.

Qism dasturning turlari.

- **Funksiya** – ma'lum bir vazifani bajaruvchi, qandaydir nomga ega, bir yoki bir necha qiymatni qabul qiluvchi, ishni tugatganidan keyin esa asosiy dasturga bir yoki bir necha natija qiymatlarni qaytaruvchi qism dastur.
- **Protsedura** – funksiyaga o'xshash ko'p marta foydalanilishi mumkin bo'lgan qism dastur bo'lib, yagona farqli jihati hech qanday qiymatni qaytarmaydi. Python dasturlash tilining har xil masalalarni yechishga mo'ljallangan bir necha foydali standart funksiyalari mavjud.

Standart funksiyalar.

print() – foydalanuvchi uchun ma'lumotlarni chiqaradi. Masalan, turli ma'lumotlar va hisoblash natijalarini.

input() – print() funksiyasining zidi, foydalanuvchilar kiritgan ma'lumotlarni dasturga uzatadi.

randint() – tasodifiy sonni chiqaradi. Masalan, dasturda tasodifiy son kerak bo'lib qolganda ishlatiladi.

Funksiyani e'lon qilish va chaqirish

Har bir yaratilgan qism dasturga, xususan, funksiya hamda protseduraga albatta nom berish kerak va bu nom Pythonida define (ing. define – aniqlash) so'zidan olingan def kalit so'zi bilan boshlanadi.

Funksiyalar def zahira so'zi orqali aniqlanadi. Bu so'zdan so'ng funksiya nomi, undan so'ng qavs va shu qavs ichida bir necha o'zgaruvchilarni ko'rsatish mumkin bo'ladi va oxirida ikki nuqta (:) yoziladi. Shulardan so'ng funksiyaning tashkil qiluvchi buyruqlar bloki yoziladi.

Sintaksisi:

def funksiya_nomi ([parametrlar ro'yxati]):
 buyruqlar_bloki

bu yerda:

def – funksiyaning e'lon qiluvchi kalit so'z.

funksiya_nomi – funksiya nomi.

parametrlar ro'yxati – ushbu ro'yxat bir necha parametrdan iborat bo'lishi mumkin va ular vergul bilan ajratib yoziladi.

buyruqlar_bloki – funksiya tanasi boshqa operatorlar kabi bitta xat boshi tashlab yozilishi shart.

Funksiya nomi orqali chaqirilganda uning tarkibidagi buyruqlar ketma-ketligi bajariladi. Shundan so'ng dastur funksiya chaqirilgan satrga qaytadi va shu satrdan keyingi buyruqlarga o'tadi.

Misol. Xabarni chiqarish.

```
def welcome():  
    msg='Xayrli kun! '  
    return msg  
  
print(welcome())  
  
Xayrli kun!
```

salomlashish nomli funksiya e'lon qilindi **xabar** o'zgaruvchisiga qiymat berish. Funksiyaning vazifasi – xabar o'zgaruvchisi qiymatini qaytarish. Funksiyani chaqirib, ekranga chiqarish.

Protsedurani e'lon qilish

```
def msg():
    print('Bugun soat 14.00 da ota-
          onalar majlisi!')

>>> msg()
Bugun soat 14.00 da ota-onalar majlisi!
```

Pythonida protseduralar deyarli funksiyalardek yoziladi. Farqi shundaki, protseduralar hech qanday qiymatni qaytarmaydi. Quyida protseduraga misol

keltirilgan:

Funksiyaga qiymat uzatish.

Funksiyaga qayta ishlashi uchun qiymatlar berish mumkin.

Misol. Aylananing radiusi kiritilganda, uning uzunligini topish dasturini tuzing.

```
def circle(r):
    PI=3.14
    len=2*PI*r
    return len
radius = int(input('Aylana radiusi: '))
uz= circle(radius)
print('Aylananing uzunligi: ', uz)

Aylananing radiusi: 4
Aylananing uzunligi: 25.136
```

aylana nomli funksiya e'lon qilindi, uning qabul qiluvchi qiymati – r. Aylananing uzunligi hisoblandi. Funksiya aylana uzunligini qaytaradi. Foydalanuvchi tomonidan kiritilgan radiusni butun qiymatga o'zgartirish. Aylana nomli funksiya chaqirilmoqda. Aylana uzunligi chiqarilmoqda.

Rekursiya

Funksiyaning o'zini o'zi chaqirishiga rekursiya deyiladi va bunday funksiyalar rekursiv funksiyalar deb ataladi.

Rekursiv funksiyalar dasturlashning kuchli mexanizmi hisoblanadi, lekin ular har doim ham samarali emas. Chunki aksariyat hollarda xatolarga yo'l qo'yadi. Xatolar ichidan eng ko'p tarqalgani – cheksiz rekursiya. Unda funksiyaning chaqiruv zanjiri cheksiz bo'lib, kompyuter bo'sh xotirasi tugamaguncha davom etaveradi. Cheksiz rekursiya ro'y berishining sabablari:

– rekursiyada shartni noto'g'ri qo'llash. Masalan, faktorialni hisoblashda if

$n==0$ ni unutib qo'ysak, `factorial(0)` funksiyasi `factorial(-1)`ni, `factorial(-1)` funksiyasi esa `factorial(-2)` va hokazolarni chaqiradi;

– rekursiv funksiyani noto'g'ri parametr bilan chaqirish. Masalan, `factorial(n)` funksiya `factorial(n)`ni chaqirsa, yana cheksiz zanjir yuzaga keladi.

Shu boisdan, rekursiv funksiyani yaratishda rekursiyani yakunlash sharti hamda rekursiyani qachon, qanday tugatish haqida o'ylab ko'rish lozim.

Har bir o'zgaruvchi e'lon qilinganda, u dasturda qandaydir muhim vazifani bajaradi va dasturning qaysidir elementlari bilan bog'lanadi. Funksiya ichida e'lon qilingan o'zgaruvchilar (lokal o'zgaruvchilar) va funksiya tashqarisida e'lon qilingan o'zgaruvchilar (global o'zgaruvchilar)ning ahamiyatlari ham turlicha.

Lokal o'zgaruvchilar.

Tayanch tushunchalar Lokal o'zgaruvchilar faqatgina funksiya ichida ahamiyatga ega bo'lib, asosiy dastur va boshqa funksiyalar ichida ishlamaydi. Shuning uchun funksiya tashqarisida lokal o'zgaruvchiga murojaat etilganda, xatolik ro'y bergani haqida xabar chiqariladi.

Lokal o'zgaruvchilar – bu o'zlari e'lon qilingan qism dasturda faol bo'ladigan o'zgaruvchilar. Ularni muayyan funksiya doirasida qo'llash mumkin, shu sababli faqat shu funksiya doirasidagina amal qiladi. Lokal o'zgaruvchilardan foydalanish o'zgaruvchining qiymati dasturning boshqa qismlarida tasodifan o'zgarib qolishi xavfini kamaytiradi.

Global o'zgaruvchilar.

Global o'zgaruvchilar butun dastur davomida faol bo'ladigan o'zgaruvchilardir. Ular qism dasturdan tashqarida, ya'ni asosiy dasturda e'lon qilinadi. Odatda, ularni modullarni import qilgandan keyin, kodning boshlanishida e'lon qilish kerak. Ularni odatiy o'zgaruvchilar kabi chaqirish mumkin.

<pre>>>> b=5</pre>	b global o'zgaruvchi e'lon qilindi va 5 qiymat
<pre>>>> def val2():</pre>	berildi.
<pre> print (b)</pre>	val2 nomli funksiya e'lon qilindi.
<pre>>>> val2()</pre>	Funksiya ichida b ning qiymatini ekranga chiqarish

	buyrug'i berildi.
5	Asosiy dasturda val2 nomli funksiya chaqirilganda, ekranga b ning qiymatini chiqaradi.
>>> print (b) 5	b global o'zgaruvchiga dasturning ixtiyoriy qismida: asosiy dasturda ham, qism dasturda ham murojaat qilish mumkin va u natijani chiqaradi.

Global o'zgaruvchining qiymatini o'zgartirish.

Agar funksiya global o'zgaruvchining qiymatini o'zgartirishga to'g'ri kelsa, u holda funksiya "global" kalit so'zi yordamida o'zgaruvchini yana qayta e'lon qilish zarur.

>>> b=5 >>> def val3(): global b b=b+1 print(b) >>> val3()	b global o'zgaruvchi e'lon qilindi va 5 qiymat berildi. val3 nomli funksiya e'lon qilindi. b o'zgaruvchining qiymatini o'zgartirish va uni butun dasturda ahamiyatli qilish uchun u funksiya ichida global sifatida qayta e'lon qilinadi. Funksiya ichida b ning qiymatini ekranga chiqarish buyrug'i berildi.
6	Asosiy dasturda val3 nomli funksiya chaqirilganda, ekranga b ning qiymatini chiqaradi.
>>> print(b) 6	b global o'zgaruvchiga asosiy dasturda murojaat qilinganda ham natija sifatida 5 ni emas 6 ni chiqaradi, sababi u funksiya ichida global sifatida e'lon qilingan.

O'zgaruvchi funksiya parametr sifatida.

Agar o'zgaruvchi funksiya parametr sifatida ishlatilsa, uning qiymati lokal o'zgaruvchining yangi qiymati sifatida qabul qilinadi.

>>> def val4(d):	d ning qiymati val4 nomli funksiya qaysi qiymat bilan chaqirilsa, shu qiymatga teng, ya'ni c ga teng.
print(d) d=100	Funksiya ichida parametr sifatida kelgan d ning qiymatini ekranga chiqarish buyrug'i berildi

print(d)	Lokal o'zgaruvchi d ga yangi qiymat berildi.
>>>c=200	d lokal o'zgaruvchining qiymati ekranga chiqarildi
>>>val4(c)	Global o'zgaruvchi c e'lon qilindi.
200 100	Asosiy dasturda val4 nomli funksiya c parametr bilan chaqirilganda, birinchi parametr sifatida kelgan 200 ni, keyin esa lokal o'zgaruvchiga yangi berilgan qiymat 100 ni chiqaradi.

Xulosa

Maqola Python dasturlash tilida funksiyalar va protseduralar tushunchalarini chuqur yoritib, ularning dasturdagi ahamiyati va ishlatilish usullarini batafsil tushuntiradi. Funksiyalar kodni modul va qayta ishlatiladigan bo'laklarga ajratib, takrorlanadigan kodni qisqartirishda muhim rol o'ynaydi. Parametrlar va argumentlar yordamida funksiyalarga moslashtirilgan qiymatlar berish imkoniyati ko'rib chiqiladi. Shuningdek, lokal va global o'zgaruvchilarni boshqarish, ularning farqlari va dasturdagi ta'siri haqida ma'lumot beriladi. Rekursiya tushunchasi va uning xavflari ham muhokama qilinib, samarali rekursiv funksiyalar yaratish uchun shartlarni to'g'ri belgilash zarurligi ta'kidlanadi. Maqolada Pythonning standart funksiyalari misollar orqali ko'rsatilgan. Umuman olganda, maqola Python dasturlashda funksiyalar va protseduralarni to'g'ri tushunish va qo'llash uchun mustahkam nazariy asos yaratadi hamda amaliy misollar bilan boyitilgan.

FOYDALANILGAN ADABIYOTLAR.

1. Informatik va axborot texnologiyalari: umumiy o'rta ta'lim maktablarining 9-sinflar uchun darslik M. R. Faysiyeva, D. M. Sayfurov, N. S. Xaytullayeva – Toshkent: Tasvir, 2020 – 112b
2. Boltayev B., Azamatov A., Asqarov A., Sodiqov M., Azamatova G. Informatika va hisoblash texnikasi asoslari. Umumiy o'rta ta'lim maktablarining 9-sinfi uchun darslik. Toshkent: "Cho'lpon" nomidagi NMIU, 2015. – 160 b.
3. Chris Roffey. Computer science. Programming book for Python. – USA: Cambridge university press. 2017, – p. 204
4. Chris Roffey. Python basics. Coding club. Level 1,2. – USA: Cambridge university press. 2012, – p. 85

5. Eric Matthes. Python crash course: a hands-on, project-based introduction to programming. – San-Francisco: No Starch Press, 2015. – p. 562
6. Matt Harrison. Illustrated guide to Python 3. 2017, – p. 267
7. Dan Bader. Python tricks the book. Anja Pircher Design, 2017, – p. 299
8. Jamie Chan. Learn python in one day and learn it well. – p. 132