Sh.B.Axmedova, D.E.Toshtemirov, M.A.Ahtamqulov

DASTURLASH ASOSLARI

O‘ZBEKISTON RESPUBLIKASI
MAKTABGACHA VA MAKTAB TA’LIMI VAZIRLIGI
ANDIJON VILOYATI PEDAGOGIK MAHORAT MARKAZI

O‘ZBEKISTON RESPUBLIKASI
OLIY TA’LIM, FAN VA INNOVATSIYALAR VAZIRLIGI
GULISTON DAVLAT UNIVERSITETI

Sh.B.Axmedova, D.E.Toshtemirov, M.A.Ahtamqulov

DASTURLASH ASOSLARI

O‘QUV QO‘LLANMA

Andijon – 2025

UO‘K 004.42(075.8)
KBK 32.973-018.2ya73
A90

Sh.B.Axmedova, D.E.Toshtemirov, M.A.Ahtamqulov. Dasturlash asoslari. [O‘quv qo‘llanma] – “Omadbek print number one” MCHJ. Andijon: 2025. – 127 b.

Mazkur o‘quv qo‘llanma “Dasturlash asoslari” fani bo‘yicha Oliy ta’lim fan, va innovatsiyalar vazirligi tomonidan tasdiqlangan o‘quv dasturida ko‘rsatilgan mavzular asosida ishlab chiqildi. Unda PYTHON dasturlash tilidan foydalanib har xil turdagi murakkab jarayonlarning dasturini tuzish oid tavsiyalar o‘rin olgan. “Dasturlash asoslari” nomli o‘quv qo‘llanma 60540100-Matematika bakalavr yo‘nalishi talabalariga mo‘ljallangan.

Ushbu o‘quv qo‘llanma AVPMM O’quv-umetodik Kengashining 2025-yil “____” _____dagi № _____ - sonli qarori bilan chop etishga tavsiya etilgan.

Taqrizchilar: **B.S.Abdullaeva,** O‘zMPU, F.d., professor

J.D.Saidov, GulDU, p.f.f.d., PhD., dots. v.b.

M.A.Kuychiyeva, AVPMM, kaf.mud., dots., prof. v.b.

ISBN : 978-9910-11-449-6

1537

© Sh.B.Axmedova,
D.E.Toshtemirov, M.A.Ahtamqulov
© Omadbek print number one

MUNDARIJA

SO‘Z BOSHI 7

KIRISH 8

I BOB. DASTURLASHNING NAZARIY ASOSLARI VA PYTHON DASTURLASH TILI 9

1.1-§. Algoritmning tasvirlash usullari. Algoritmning xossalari. Chiziqli, tarmoqlanuvchi, takrorlanuvchi algoritmlar. 9

1.2-§. Dasturlash tillarining klassifikatsiyasi va python dasturlash tili. 21

1.3-§. Python dasturlash tilining IDLE rejimida interaktiv dastur yaratish. Python dasturlash tilida faylli dastur yaratish. 27

1.4-§. Python dasturlash tilining standart kutubxonalari va tashqi kutubxonasini o‘rnatish. 36

1.5-§. Python dasturlash tili tarkibidagi matematik funksiyalar va ifodalar. 47

II BOB. PYTHON DASTURLASH TILIDA CHIZIQLI, TARMOQLANUVCHI VA TAKRORLANUVCHI JARAYONLARNI DASTURLASH. 54

2.1-§. Dasturlash tili tarkibidagi o‘zgaruvchi va o‘zgarmaslarni e’lon qilish. 54

2.2-§. Python dasturlash tilida chiziqli jarayonlarni dasturlash. 63

2.3-§. Python dasturlash tilida tarmoqlanuvchi jarayonlarni dasturlash. 70

2.4-§. Takrorlanuvchi jarayonlarni dasturlash, for(sikl) operatori va uning umumiy ko‘rinishi. 78

III BOB. PYTHON DASTURLASH TILIDA MURAKKAB TURLAR 88

3.1-§. Ro‘yxatlar. 88

3.2-§. Kortejlar. 93

3.3-§. Python dasturlash tilida funksiyalar. 96

3.4-§. Fayllar ustida amallar bajarish. 103

3.5-§. Pythonda geometrik shakllar chizish, chiziq qalinligi va harakatlanish vaqti. 109

Xulosa 120

Glossariy 121

Foydalanilgan adabiyotlar ro‘yxati 124

ОГЛАВЛЕНИЕ

НАЧАЛО СЛОВА……6
ВВЕДЕНИЕ……..……7
ГЛАВА I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОГРАММИРОВАНИЯ И ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON……...……8
§ 1.1. Методы описания алгоритмов. Свойства алгоритма. Линейные, ветвящиеся, итерационные алгоритмы……8
§ 1.2. Классификация языков программирования и язык программирования Python…….……..20
§ 1.3. Создание интерактивной программы в режиме IDLE языка программирования Python. Создание файловой программы на языке программирования Python……. 26
§ 1.4. Установка стандартных библиотек и внешней библиотеки языка программирования Python……. 35
§ 1.5. Математические функции и выражения на языке программирования Python……. 46
ГЛАВА II. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ, РАЗВЕТВЛЕННЫХ И ПОВТОРЯЮЩИХСЯ ПРОЦЕССОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON……...…….. 53
§ 2.1. Объявление переменных и констант на языке программирования. ……...53
§ 2.2. Программирование линейных процессов на языке программирования Python…….…. 62
§ 2.3. Программирование ветвящихся процессов на языке программирования Python…….…. 69
§ 2.4. Программирование циклов, оператор for(loop) и его обзор……. 77
ГЛАВА III. СЛОЖНЫЕ ТИПЫ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON……. 87
§ 3.1. Списки…… 87
§ 3.2. Шествия…… ….92
§ 3.3. Функции на языке программирования Python……... 95
§ 3.4. Выполнение действий над файлами……..……... 102
§ 3.5. Рисование геометрических фигур, толщины линий и времени движения на Python……...109
Вывод…..…….....119
Глоссарий…...……...120
Список использованной литературы……..123

ABLE OF CONTENTS

PREFACE……..6
INTRODUCTION……7
CHAPTER I. THEORETICAL FOUNDATIONS OF PROGRAMMING AND THE PYTHON PROGRAMMING LANGUAGE……...8
1.1-§. Methods of describing an algorithm. Properties of an algorithm. Linear, branching, iterative algorithms……....8
1.2-§. Classification of programming languages and the Python programming language……20
1.3-§. Creating an interactive program in the IDLE mode of the Python programming language. Creating a file program in the Python programming language……26
1.4-§. Installing the standard libraries and external libraries of the Python programming language……..35
1.5-§. Mathematical functions and expressions in the Python programming language……46
CHAPTER II. PROGRAMMING LINEAR, NETWORK, AND REPEATABLE PROCESSES IN THE PYTHON PROGRAMMING LANGUAGE……53
2.1-§. Declaring variables and invariants in the programming language……..53
2.2-§. Programming linear processes in the Python programming language…62
2.3-§. Programming branching processes in Python……69
2.4-§. Programming repetitive processes, the for(loop) operator and its general view……...77
CHAPTER III. COMPLEX TYPES IN PYTHON……..87
3.1-§. Lists……87
3.2-§. Tuples……..92
3.3-§. Functions in Python……95
3.4-§. Performing operations on files……..102
3.5-§. Drawing geometric shapes in Python, line thickness and execution time……..109
Conclusion ……119
Glossary ……120
List of references ……...123

So‘z boshi

Hozirgi vaqtda barcha sohalarni axborot texnologiyalarisiz tasavvur qilib bo‘lmaydi. Jamiyatning barcha sohalariga axborot texnologiyalari jadallik bilan kirib kelayotgan bir vaqtda, ta’lim sohasida ham axborot texnologiyalarining o‘rni keskin oshib bormoqda. Bugungi kunga kelib, deyarli barcha sohalarda o‘zining ichki jarayonlari va funksiyalarini qayta tashkil etish, ularni avtomatlashtirish, shu bilan dasturchilar va ishlab chiquvchilarga bo‘lgan talabni oshirayotgan ko‘plab kompaniyalarni kuzatishimiz mumkin. Lekin ularni qayerdan topish mumkin? Va zamon bilan hamnafas bo‘lib IT sohasida yaxshi mutaxassis bo‘lish uchun qayerda o‘qish va o‘rganish kerak? Bu borada so‘nggi paytlarda mamlakatimizda sanog‘i kun sayin oshib borayotgan kurslar yordam berishi mumkin. Dasturlash asoslari — bu zamonaviy dunyoda har bir inson uchun zarur bo‘lgan bilimlardan biridir. Bizning hayotimizning ko‘plab jabhalarida dasturlash va kod yozish muhim rol o‘ynaydi, shuning uchun bu faoliyatga qiziqish va asosiy tushunchalarni o‘rganish juda muhimdir.

Dasturlash, yoki kompyuter dasturlarini yaratish jarayoni, zamonaviy texnologiyalar dunyosida muhim o‘rin tutadi. Oddiy til bilan aytganda, dasturlash – bu kompyuter va boshqa qurilmalarga qanday qilib ma'lum vazifalarni bajarishni buyurish. Bu jarayon dasturlash tillari yordamida amalga oshiriladi, masalan, Python, Java, C++ kabi tillar. Har bir dasturlash tili o‘ziga xos sintaksis va qoidalar to‘plamiga ega, shuning uchun dasturchilar o‘z ehtiyojlariga mos tilni tanlashlari kerak.

Dasturlash orqali yaratilgan dasturlar oddiy vazifalardan tortib, murakkab tizimlargacha bo‘lgan keng doiradagi ilovalarni o‘z ichiga oladi. Misol uchun, mobil ilovalar, veb-saytlar, o‘yinlar va hatto sun'iy intellektga asoslangan tizimlar ham dasturlash orqali yaratiladi. Dasturlash nafaqat texnik ko‘nikmalarga ega bo‘lishni talab qiladi, balki mantiqiy fikrlash, muammoni hal qilish va yaratuvchanlikni ham rivojlantiradi.

KIRISH

Ushbu o‘quv qo‘llanma talabalarga dasturlashni o‘rganish uchun, qanday dastur tuzishni tez va samarali o‘rganish imkonini beradi. Qo‘llanmada python dasturlash tili tarkibidagi amallar, operatorlar, funksiyalar, murakkab turlar, fayllar va grafiklar bo‘yicha ma’lumotlarni qayta ishlash usullari ko‘rsatib o‘tilgan. O‘quv qo‘llanmani tushunarli bo‘lishi uchun biz, eng sodda jarayonlardan, murakkab jarayonlarni dasturlashga tomon intildik. Python dasturlash tili tarixi uzoq tarixga bormasada, xozzirgi kunda uning rivojlanishi eng yuqori o‘rinlardan saft olgan. Python — (payton, piton) — turli sohalar uchun yuqori darajadagi umumiy maqsadli dasturlash tili. Uning dizayn falsafasi muhim chekinishdan foydalangan holda kodning o‘qilishiga urg‘u beradi. Uning til konstruksiyalari va obyektga yo‘naltirilgan yondashuvi dasturchilarga kichik va yirik loyihalar uchun aniq, mantiqiy kod yozishda yordam berishga qaratilgan. Shuningdek Python sun’iy intellekt hamda ma’lumotlar muhandisiligi sohalarining tili hisoblanadi. Python deyarli barcha platformalarda ishlay oladi, xususan Windows, Linux, Mac OS X, Palm OS, Mac OS va boshqalar shular jumlasidandir. Python Microsoft.NET platformasi uchun yozilgan realizatsiyasi ham mavjud bo‘lib, uning nomi — IronPython dasturlash muhitidir.
Python quyidagilar uchun ishlatiladi:

- veb-ishlab chiqish (server tomonida),
- dasturiy ta’minotni ishlab chiqish,
- matematik amallar,
- tizim skriptlari.

Dasturlashni o‘rganish, o‘z navbatida, yangi imkoniyatlarga yo‘l ochadi. Bugungi kunda, deyarli har bir sohada kompyuter texnologiyalari va dasturlash bilimlari talab etiladi. Shuning uchun, dasturlash nima ekanligini tushunish va uni o‘rganish – bu zamonaviy dunyoda muvaffaqiyatga erishishning birinchi qadami hisoblanadi. Bu adamni qo‘yganingizda, siz nafaqat o‘z mahoratingizni oshirasiz, balki kelajakdagi kasbiy imkoniyatlaringizni ham kengaytirasiz.

I BOB. DASTURLASHNING NAZARIY ASOSLARI VA PYTHON DASTURLASH TILI

1.1-§. Algoritmning tasvirlash usullari. Algoritmning xossalari. Chiziqli, tarmoqlanuvchi, takrorlanuvchi algoritmlar.

Reja:

1. *Algoritm;*
2. *Algoritmning tasvirlash usullari;*
3. *Algoritmning xossalari;*
4. *Algoritmning turlari.*

Tayanch so‘zlar: *Algoritm, algoritm xossalari, algoritm turlari. Chiziqli algoritmlar, tarmoqlanuvchi algoritmlar, takrorlanuvchi algoritmlar.*

Algoritm tushunchasi IX-asrda yashab ijod etgan buyuk mutaffakir bobokolonimiz Muhammad al-Xorazmiy nomi bilan uzviy bog‘liqligini tushuntirish lozim. Algoritm so‘zi al-Xorazmiyning arifmetikaga bag‘ishlangan asarining dastlabki betidagi: Dixit Algoritmi” (“dediki al-Xorazmiy” ning lotincha ifodasi) degan jumlalardan kelib chiqqan. Shu bilan birga al-Xorazmiy sanoq sistemasini yuksalishiga qo‘shgan hissasi, uning asarlari algoritm tushunchasining kirib kelishiga sabab bo‘lganini aytib o‘tish lozim.

Algoritm o‘zi nima degan savolga, u asosiy tushuncha sifatida qabul qilinganligidan, uning tavsifi beriladi, ya’ni biron-bir maqsadga erishish uchun yoki qandaydir masalani yechimi topish uchun beriladigan ko‘rsatmalar (buyruqlar)ning aniq, tushunarli, chekli hamda to‘liq tizimi tushuniladi.

Algoritm so‘zining bir qancha ta’rifi mavjud bo‘lib, ulardan ayrimlarini keltirib o‘tamiz:

- Algoritm – bu belgilaydigan cheklangan qoidalar to‘plami, muayyan vazifalar to‘plamini hal qilish bo‘yicha amallar ketma-ketligi va beshta muhum xossaga ega: aniqlilik, tushunarlilik, kiritish, chiqarish, samaradorlik (D.E.Knut).
- Algoritm – bu qat’iy belgilangan qoidalar asosida bajariladigan har qanday hisoblash tizimidir, bu ma’lum bir qator bosqichlardan so‘ng, aniq qo‘yilgan masalani hal qilishga olib keladi (A.Kolmogorov).
- Algoritm – bu har xil boshlang‘ich ma’lumotlardan kerakli natijaga o‘tadigan hisoblash jarayonini belgilaydigan aniq ketma-ketlik (A.Markov).

Ta’rif: Algoritm – bu biror bir ko‘zlangan maqsadga erishish yoki muammoni yechimi topish uchun ijrochi tomonidan beriladigan chekli sondagi buyruqlar ketma-ketligi.

Masalan, bizga tomonlari a va b sonlarga teng bo‘lgan to‘g‘ri turtburchakning yuzini topish masalasi berilgan bo‘lsin, uning algoritmini tuzish tartibi ko‘rib chiqamiz.

1. a va b tomonlar kiritilsin;
2. kiritilgan kesmalar yordamida S=a*b hisoblansin.

Demak, masalani yechish ketma-ketligini algoritm sifatida qarash mumkin.

Algoritmning tasvirlash usullari

Algoritmning tasvirlash usullari haqida gap ketganda algoritmning berilish usullari xilma-xilligi va ular orasida eng ko‘p uchraydiganlari quydagilar ekanligini ko‘rsatib o‘tish lozim:

1. Algoritmning so‘zlar yordamida tasvirlanishi. Bu usulda algoritm ijrochi uchun beriladigan barcha buyruqlar ketma-ketligi so‘zlar orqali beriladi.

Misol. Katetlari a va b bo‘lgan to‘g‘ri burchakli uchburchakni gipotenuzasini hisoblang.

1. Boshlansin
2. a katet uzunligi kvatratga ko‘tarilsin;
3. b katet uzunligi kvatratga ko‘tarilsin;
4. natijalar qo‘shilsin va c ning kvadratiga tenglashtirilsin;
5. chiqqan natijadan ildiz chiqarilsin;
6. Tamom

2. Algoritmning formulalar yordamida tasvirlanishi. Bu usulda algoritmning har bir amali matematik formulalar yordamida ifodalanadi. Algoritm amallarini yozishda oddiy matematik yozuvlardan foydalanish mumkin[1].

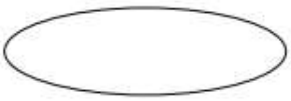
Misol: $ax^2+bx+c=0$ kvadrat tenglama yechimini toping.

1. Boshlansin;
2. Aniqlansin a, b, c;
3. Hisoblansin $D:=b^2-4ac$;
4. Agar $D<0$ bo‘lsa chiqarilsin “Yechimi yo‘q”, 7 ga o‘tilsin;

5. Hisoblansin $x_1 := \frac{-b+\sqrt{D}}{2a}$, $x_2 := \frac{-b-\sqrt{D}}{2a}$;
6. Chiqarilsin x_1, x_2;
7. Ish tugatilsin.

3. Algoritmning jadval ko‘rinishida tasvirlanishi – bu usulda turli matematik jadvallar, loteriya yutuqlari jadvali, funksiyalar qiymatlari jadvallari bunnga misol bo‘la oladi.

4. Algoritmning grafik shaklida tasvirlanishi algoritmlar maxsus geometrik figuralar yordamida tasvirlanadi va bu grafik ko‘rinishi blok-sxema deyiladi. Algoritmning blok-sxema ko‘rinishida tasvirlanishi beriladigan buyruqlar qandaydir shakllar orqali ifodalanadi. Algoritmning blok-sxemalar yordamida tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat

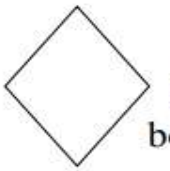


– Ellips u algoritmning boshlanishi yoki tugallashini belgilaydi;

– To‘g‘ri burchakli to‘rtburchak, qiymat berish yoki tegishli ko‘rsatmalarni bajarish jarayonini belgilaydi;

– Parallelogramm, ma’lumotlarni kiritish yoki chiqarishni belgilaydi.

– Yordamchi algoritmga murojatni belgilaydi.

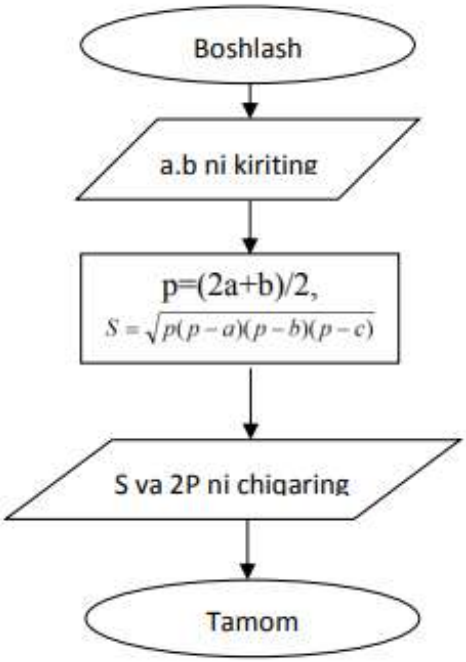


– Romb, shart tekshirishni belgilaydi va shart bajarilsa "ha", tarmoq bo‘yicha, aks holda "yo‘q”-tarmog‘i bo‘yicha amallar bajarilishini ta’minlaydi.

- → Strelka - amallar ketma-ketligining bajarilish yo‘nalishini ko‘rsatadi.

Yuqorida ko‘rilgan algoritmlarning tasvirlash usullarining asosiy maqsadi - qo‘yilgan masalani yechish uchun zarur bo‘lgan amallar ketmaketligining eng qulay holatini aniqlash va shu bilan foydalanuvchi tomonidan programma yozishni yanada osonlashtirishdan iborat. Aslida programma ham algoritmning boshqa bir ko‘rinishi bo‘lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo‘ljallangan[2].

Misol. Teng yonli uchburchakning tomonlari berilganda uning yuzi va perimetrini topish algoritm blok-sxemasini yozing.



Algoritmning xossalari

Algoritmning asosiy xossalari sifatida quydagilarni keltirish mumkin:

1-xossa. Diskretlilik, ya’ni algoritmni chekli sondagi oddiy ko‘rsatmalar ketma-ketligi shaklida ifodalash mumkin.

2-xossa. Tushinarlilik, ya’ni ijrochiga tavsiya etilayotgan ko‘rsatmalar uning uchun tushunarli bo‘lishi shart,aks holda ijrochi oddiy amalni ham bajara olmay qolishi mumkin. Har bir ijrochining bajara olishi mumkin bo‘lgan ko‘rsatmalar tizimi mavjud.

3-xossa. Aniqlilik, ya’ni ijrochiga berilayotgan buyruqlar aniq mazmunda bo‘lishi lozim hamda faqat algoritmda ko‘rsatilgan tartibda bajarilishi shart.

4-xossa. Ommaviylik, ya’ni har bir algoritm mazmuniga ko‘ra bir turdagi masalalarning barchas uchun yaroqli bo‘lishi lozim. Masalan,

ikki oddiy kasr umumiy maxrajini topish algoritmi har qanday kasrlar umumiy maxrajini topish uchun ishlatiladi.

5-xossa. Natijaviylik, ya’ni har bir algoritm chekli sondagi qadamlardan so‘ng albatta natija berishi lozim.

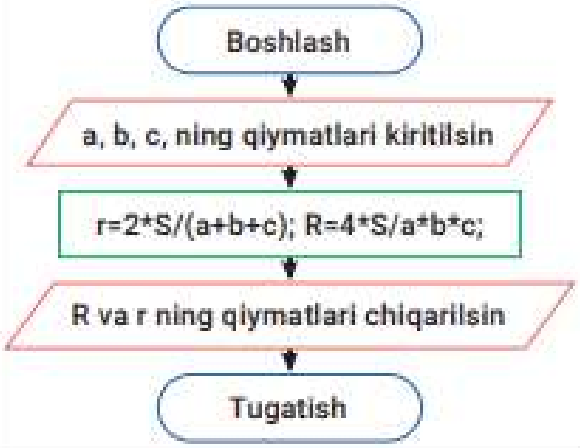
Bu xossalar mohiyatini o‘rganish va konkret algoritmlar uchun qarab chiqish talabalarning xossalar mazmunini bilib olishlariga ko‘maklashadi.

Algoritmning turlari

Algoritmlar turli xil vazifalarni bajarish uchun qo‘llaniladigan ko‘rsatmalar yig‘indisidir. Ular qo‘yilgan maqsadga erishish uchun qadam-baqadam bajariladigan jarayonlarni tasvirlaydi. Algoritmlar bir necha turga bo‘linadi, ular orasida eng asosiylari quyidagilar:

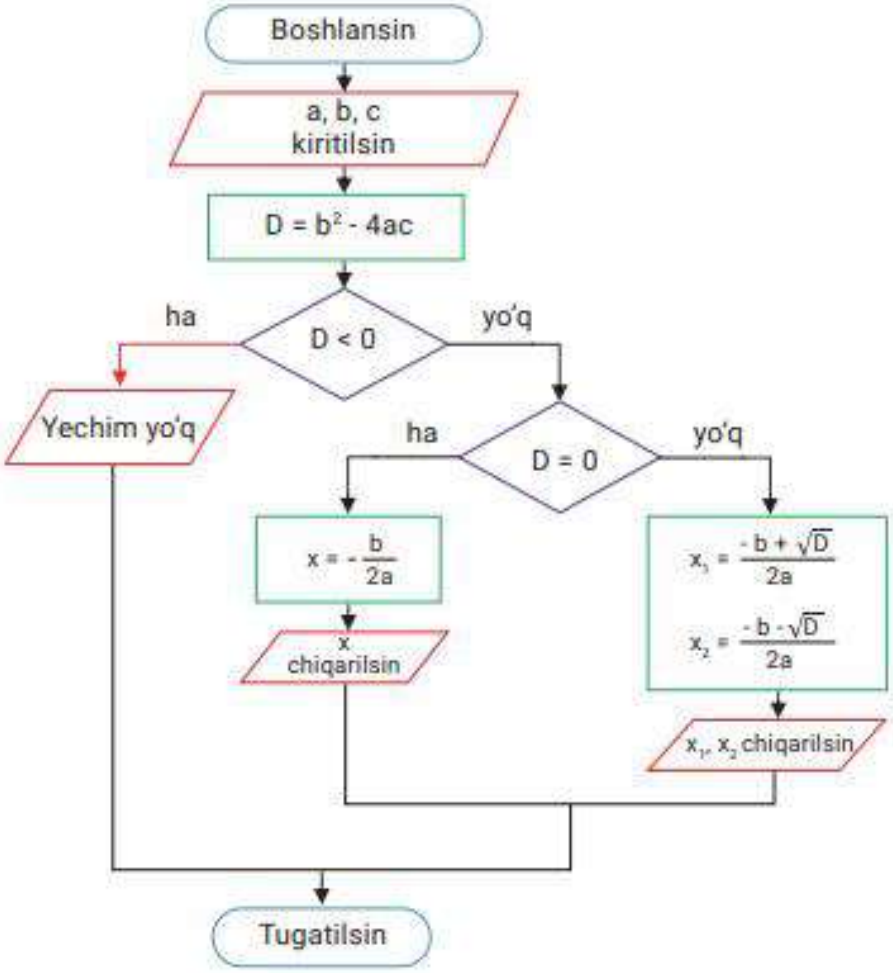
1. **Chiziqli algoritmlar** – bu algoritmlar ketma-ketlikda, hech qanday shartsiz bajariladi. Barcha qadamlar birin-ketin bajariladi. Misol uchun, oddiy hisoblash jarayonlari yoki ma’lumot kiritish-chiqarish jarayonlari bu turga kiradi.

Misol. Uchburchak tomonlarining uzunligi bilan berilgan. Uchburchakga ichki va tashqi chizilgan aylanalar radiuslari va uzunliklari hisoblansin. Ichki chizilgan aylana radiusi $r=2S/(a+b+c)$, tashqi chizilgan aylananing radiusi esa $R=4S/abc$ formulalar orqali hisoblanadi. Bu yerda S – uchburchakning yuzi, a, b, c – uchburchak tomonlarining uzunliklari.

1) boshlansin; 2) *a, b, c* larning qiymatlari aniqlansin; 3) ichki chizilgan aylananing radiusi: *r=2S/(a+b+c)* hisoblansin; 4) tashqi chizilgan aylananing radiusi: *R=4S/abc* hisoblansin; 5) *r* va *R* qiymatlari chop etilsin; 6) tugatilsin.	

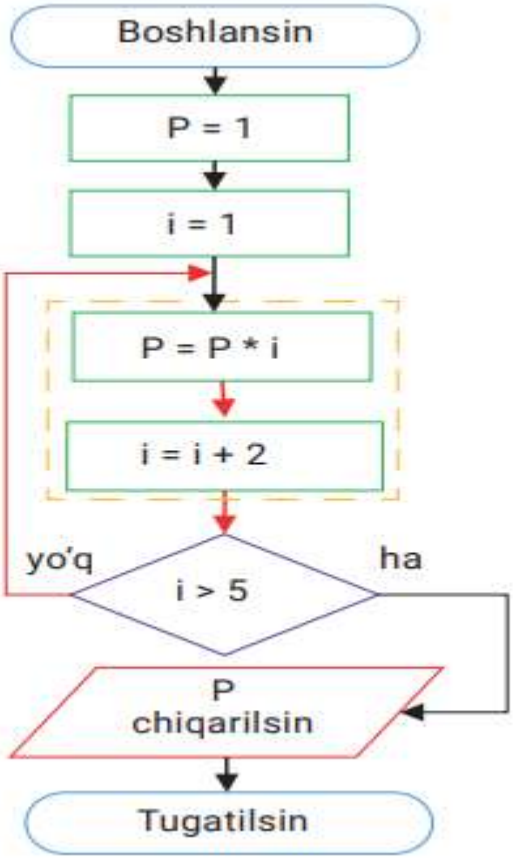
2. **Tarmoqlanuvchi algoritmlar** – bu algoritmlar bajarilishi uchun ma’lum shartlar talab qilinadi. Agar shart bajarilsa, bir yo‘lga o‘tiladi, agar bajarilmasa, boshqa yo‘lga. Misol uchun, sonlar orasidan kattasini topish algoritmi tarmoqlanuvchi algoritmdir.

Misol. Berilgan $a*x^2+b*x+c=0$ kvadrat tenglamani ildizlarini topish algoritmini tuzing



3. **Takrorlanuvchi (siklik) algoritmlar** – ma’lum bir shart asosida yoki parametrlar o‘zgarishi bilan jarayonlar bir necha marta takrorlanadigan algoritmlardir. Misol uchun, 1 dan 100 gacha bo‘lgan sonlar yig‘indisini hisoblashda takrorlanuvchi algoritmdan foydalaniladi.

Misol. 1 dan 5 gacha bo‘lgan sonlar ko‘paytmasini hisoblash algoritmini tuzing. Matematik modeli: P= 1· 2· 3· 4· 5=120



Algoritmlarni blok-sxemalarda tasvirlash ko‘pincha qulay bo‘ladi. Har xil shakllar yordamida kiritish, hisoblash va shartlarni tekshirish kabi jarayonlarni ko‘rsatish mumkin. Bu turdagi grafik tasvirlash algoritmlarni vizual tushunishni osonlashtiradi.

Nazariy savollar

1. Algoritm so‘zining ma’nosi
2. Algoritmning ta’rifi
3. Algoritm necha xil usulda tasvirlanish mumkin
4. Algoritmning xossalari va ularning ta’riflari
5. Algoritm turlari
6. Chiziqli algoritm deb nimaga aytiladi?
7. Tarmoqlanuvchi algoritm deb nimaga aytiladi?
8. Takrorlanuvchi algoritm deb nimaga aytiladi?

Mustaqil ishlash uchun topshiriqlar

1. Radiusi r bo‘lgan aylanaga tashqi chizilgan muntazam n-burchakning perimetrini toping.

2. Aylananing yuzasi S berilgan. Uning diametri D va uzunligi L aniqlansin. $L=2\cdot pi\cdot R$, $S=pi\cdot R^2$

3. Sonlar o‘qida ikkita nuqta orasidagi masofa aniqlansin. $|x_2-x_1|$

4. To‘g‘ri burchakli uchburchakning gipotenuzasi hamda bitta kateti berilgan bo‘lsin. Uning yuzi va ichki chizilgan aylanasining radiusini toping

5. Aylananing uzunligi l bo‘lsin. Shu aylana bilan chegaralan-gan doira yuzini aniqlang.

6. Agar halqaning ichki radiusi 20, tashqi radiusi esa undan x miqdorga katta bo‘lsa, halqaning yuzini toping.

7. a va d haqiqiy sonlar hamda n natural soni berilgan bo‘lsin. Quyidagi a, a+d, a+2d, ... , a+(n-1)d arifmetik progressiyaning dastlabki n ta hadining yig‘indisi topilsin.

8. Asoslari a va b, a asosga yopishgan burchagi α bo‘lgan teng yonli trapetsiya yuzini toping.

9. Uchburchakning tomonlari berilgan. Shu uchburchakning:

a) balandliklari topilsin;

b) medianalari topilsin;

c) bissektrisalari topilsin;

d) ichki va tashqi chizilgan aylanalarining uzunliklari topilsin.

10. Koordinatalari (x1, y1) va (x2, y2) bo‘lgan ikki nuqta orasidagi masofani toping.

11. a, b va c haqiqiy sonlar berilgan bo‘lsin. Tomonlari shu sonlarga teng bo‘lgan uchburchakni yasash mumkin. Uchburchak burchaklarini aniqlang.

12. Ikkita haqiqiy son berilgan. Agar ularning birinchisi ikkinchisidan kichik bo‘lsa, uni nol bilan, aks holda berilgan sonlarning o‘rta arifmetigi bilan almashtiring.

13. Ikkita haqiqiy son berilgan. Bu sonlarning kichigini yarim yig‘indisi bilan, kattasini esa ko‘paytmasi bilan almashtiring

14. Uchta o‘zaro har xil sonlarning yig‘indisi birdan kichik bo‘lsa, berilgan sonlarning eng kichigi, aks holda eng kattasi topilsin.

15. Musbat va haqiqiy a, b, c va d sonlari berilgan bo‘lsin. Tomonlari a va b bo‘lgan to‘g‘ri to‘rtburchak ichiga uning tomonlariga parallel yoki perpendikulyar qilib tomonlari c va d bo‘lgan to‘g‘ri to‘rtburchakni joylash mumkinmi ?

16. x haqiqiy son berilgan bo‘lsin. sinx, sinx 2 , sinx 3 va sinx 5 sonlari ichidan musbatlarini aniqlang.

17. a, b, c, d va e haqiqiy sonlar berilgan bo‘lsin. Shu sonlarni o‘sish tartibida eng kam taqqoslashlar bilan tartiblang.

18. N natural soni berilgan bo‘lsin.

a) Bu sonda qancha raqam qatnashgan ?

b) N soning raqamlari yig‘indisi nimaga teng ?

c) N sonini yozishda 3 raqami qatnashganmi ?

d) N sonidagi birinchi va oxirgi raqamlarni aniqlang.

19. N va M natural sonlari berilgan bo‘lsin. N sonining oxirgi M ta raqamlari yig‘indisini toping.

20. N va M natural sonlari berilgan bo‘lsin. N/M kasrni qisqarmaydigan ko‘rinishga keltiring.

21. N>1 butun son berilgan bo‘lsin. N dan katta bo‘lgan va 2^r ko‘rinishida bo‘lgan butun sonlarning eng kichigini toping.

1.2-§. Dasturlash tillarining klassifikatsiyasi va Python dasturlash tili.

Reja:

1. *Dasturlash tillarining klassifikatsiyasi;*
2. *Pyton dasturlash tili va uning imkoniyatlari.*

Tayanch so‘zlar: *axborot tizimi, quyi darajadagi til, o‘rta darajadagi til, yuqori darajadagi til, komplyator, translyator, interpretator.*

Dasturlash tillarining klassifikatsiyasi

Axborot tizimi va axborot texnologiyalarining avtomatlashtirilgan elementlarini qo‘llash va avtomatlashtirish asosida yangi axborot texnologiyalarini yaratish avtomatlashtirish tizimlarini loyihalashtiruvchilarning asosiy vazifalaridan biridir.

Avtomatlashtirilgan tizimlarni yaratish uchun albatta birinchi navbatda muammo obektini infologik yoki diskretli modelini qurish dolzarb hisoblanadi. Infologik yoki diskretli modelni muammo obektiga qarab algoritmlash tillarini qaysi biri asosida yaratish kerakligini tanlab olinish kerak. Elektron hisoblash mashinalarini birinchi avlodlari yaratilishi bilan algoritmlash tillarining rivojlanishi ham boshlandi. Avval algoritm tuzuvchi mutaxassislar eng sodda mashina tilini o‘zida ifodalovchi kompyuter buyruqlari bilan ishlaganlar.

Bu buyruqlar nol va birlar ketma-ketligidan iborat matnlardan tashkil topgan edi. Keyinchalik insonlar uchun tushunarli bo‘lgan mashina buyruqlarini o‘zida saqlovchi assembler tili yaratildi. Keyinchalik FORTRAN, BASIC, PASKAL va COBOL singari yuqori darajali tillar ham paydo bo‘ldiki, bu tillar yordamida so‘z va gaplarning mantiqiy konstruksiyasidan foydalanib algoritmlash imkoniyati yaratildi. Ular buyruqlarni mashina tiliga interpretatorlar va kompilyatorlar yordamida o‘tkazar edi.

Algoritmlash tillari yaratilishi bo‘yicha uchta turga ajratiladi:

-quyi darajadagi;

-o‘rta darajadagi;

-yuqori darajadagi.

Ma’lumki, ma’lumbir masalani yechish uchun buyruqlar ketma-ketligi ya’ni algoritm algoritmlash tilida yozilayotganda kamroq buyruqlardan foydalanilsa, bunday tillar darajasi yuqoriroq hisoblanadi.

Quyi darajadagi algoritmlash tillari bevosita kompyuter qurilmalari bilan bog‘liq bo‘lib, buyruqlar ularning kodlari bilan yoziladi. Bu kabi buyruqlardan tashkil topgan algoritmlar katta hajmli bo‘lib, ularni taxrirlash katta mehnat talab qiladi. Dastlabki kompyuterlar(*ENIAK*, *MESM* va boshqalar) ana shunday tillarda ishlagan.

O‘rta darajadagi algoritmlash tillari buyruqlarida faqat raqamlar emas, balki insonlar tushunadigan ba’zi so‘zlar ishlatila boshlandi(*Assembler*).

Yuqori darajadagi algoritmlash tillari quyidagicha bosqichlarga bo‘linadi:

Algoritmik(Basic, Pascal, C va b.)

Mantiqiy(Prolog, Lisp va b.)

Obyektga mo,,ljallangan(Object Pascal, PYTHON, Java va b.)

Algoritmlash tillarida yaratilgan algoritmlar mashina tiliga *translyatorlar* yordamida o‘tkaziladi.

Translyator(translator-tarjimon) biror bir algoritmlash tilida yozilgan algoritmni mashina tiliga tarjima qiladi.

Translyatorlar ikki turda bo‘ladi:

- Kompilyatorlar(*compiler*-yig‘uvchi) biror bir algoritmlash tilida yozilgan algoritmni mashina tiliga to‘liq o‘qib olib tarjima qiladi;
- Interpretatorlar(*interpreter* - izohlovchi, og‘zaki tarjimon) biror bir algoritmlash tilida yozilgan algoritmni mashina tiliga satrma - satr tarjima qiladi.

Pyton dasturlash tili va uning imkoniyatlari

Python dasturlash tilining tarixi o‘tgan asrning 80-yillari oxirlarida boshlangan. Gido Van Rossum Python dasturlash tilini 1980-yillarda yaratgan va u til 1989 yil dekabrda Gollandiyadagi matematika va informatika laboratoriya markazida ishlab chiqilgan. Python istisno holatlarini ko‘rib chiqishga va Amoeba operatsion tizimiga ta’sir ko‘rsatishga qodir bo‘lgan ABC dasturlash tilining avlodi bo‘lgan. Van Rossum Pythonning asosiy muallifidir va u 2018 yilgacha tilni rivojlantirish bo‘yicha bir qancha ishlar olib borgan.

Van Rossum tomonidan Python 1.2 versiyasi 1995 yili matematika va informatika laboratoriya markazida ishlayotgan paytda ishlab chiqarilgan. Python dasturlash tili mukammal darajada ishlab chiqilgan dasturlash tili bo‘lib u insoniyat oldidagi muammolarni hal qilish uchun

juda mos til hisoblanadi. Python dasturlash tili, dasturlash tillarining eng keng imkoniyat doirasiga ega hisoblanadi, bu dasturlash tili boshqa dasturiy vositalarni boshqarish va ularning tarkibiy qismlarini mustaqil boshqarishni amalga oshirdi. Aslida, Python ko‘p maqsadli dasturlash tili sifatida o‘rganilishi mumkin, bu dasturlash tili yordamida bir qancha jarayonlarni dasturlash imkoni yaratiladi. Python maliy dasturiy maxsulotlar, web ilovalar va ilmiy dasturiy maxsulotlar yaratish imkonini beradi. Python tarkibida xotiradan foydalanish va ishlash talablari bo‘yicha cheklovlar mavjud emas ya’ni imkoniyatlar shu qadar kattaki, boshqa dasturlash tillari kabi ma’lumotlarni e‘lon qilish tabaqasi mavjud emas. Bu esa dastur yozish vaqti kamaytiradi va boshqarish qulayligini oshiradi.

Python dasturlash tilini bu qadar keng tarqalishining sababi juda katta miqdordagi yuqori sifatli tayyor bepul tarqatiladigan modullar mavjud va ularni siz dasturning istalgan joyidan foydalanishingiz mumkin. Tayyor modullardan foydalangan holda dasturni tuzish bir qancha optimal hisoblanadi. Dasturlash tili tarkibida fundamental algoritmlar, funksiyalar va modullar tayyor holatga keltirilgan, bunda faqatgina bu algoritm yoki funksiyalarga murojaat qilinsa yetarli siz faqat tegishli qismlarni tanlashingiz va ularni bir joyga to‘plashingiz kerak. Modullar har bir misolning boshida mavjud bo‘lgan import buyrug'i yordamida biriktiriladi. Ko‘p ishlatiladigan modullar ikkita asosiy qismga bo‘lingan:

- Python interpretatori bilan ta’minlangan standart kutubxonaning modullari (ushbu modullar doimo dastur bilan birga aktivlashadi);
- Tashqi vazifa bajaruvchi modullar, bu modullar alohida dastur tarkibiga o‘rnatish orqali amalga oshiriladi.

Python dasturlash tilining web dasturlash sohasiga ham to‘g‘ridan to‘g‘ri qo‘llanilishi mumkin. Python an’anaviy ravishda oddiy va murakkab strukturali saytlarni yaratish uchun foydalaniladi. Bu jarayoning eng keng tarqalgan vositasi Django web platformasi hisoblanadi. Bu platforma orqali bir qancha keng ommaga tarqalgan mashhur tizimlar ishlab chiqilgan, jumladan, Instagram, Mozilla va hakoza. Django juda ko‘p turli xil funktsiyalarni, shu jumladan avtomatik ravishda hosil qilinadigan ma’lumotlar bazasini yaratish imkoniyatlarini taqdim etadi. Python dasturlash tili ko‘plab mashhur o‘yinlarni ishlab chiqish uchun ishlatiladi, 2000 - yillarning birinchi yarmida Python dasturlash tili sivilizatsiyasida, to‘rtinchi o‘yinning ichki mantiqiy tuzilishini yozish uchun asosiy vosita sifatida ishlatildi.Python dasturlash tilining matematik va ilmiy hisoblash jarayonlariga keng qo‘llanilish imkoniyati yaratilgan. Python umumiy maqsadlar uchun mo‘ljallangan til bo‘lib, u matematik paketlar bilan muvaffaqiyatli moslashuvni amalga oshiradi. Python dasturlash tilining asosiy xususiyati uning kengayish imkoniyatidir. Bu esa Python uchun nafaqat C va Fortrandagi algoritmlarning ko‘p sonli kutubxonalari yozildi va moslashtirildi. Python dasturlash tilining boshqa dasturiy vositalar va paketlardan

foydalanish imkoniyati mavjud. Pythonni matematik paketga aylantirishning asosiy modullari ishlab chiqilgan.

Python dasturlash tilining eng muhim afzalliklaridan biri shundaki, uning barcha amaliy kutubxonalari va qo‘shimcha maxsus modullarining rivojlanish muhiti bepul tarqatiladi. Bu esa Python dasturlash tilini rivojlantirish vositasi bo‘lishi mumkinligini anglatadi.

Python dasturlash tili dasturlashning quyidagi sohalarrida qo‘llaniladi:

- Tizimli dasturlash;
- Grafik interfeysli dasturlarni ishlab chiqish;
- Dinamik veb-saytlarni yaratish;
- Komponentlarning integratsiyasi;
- Ma’lumotlar bazalari bilan ishlash uchun dasturlarni ishlab chiqish;
- Ilmiy hisoblash uchun dasturlarni ishlab chiqish;
- O‘yinlarni rivojlantirish.

Nazariy savollar

1. Dasturlash tillarining sinflari va ularning tarkibi?
2. Quyi, o‘rta va yuqori darajali dasturlash tillarining vazifalari?
3. Quyi, o‘rta va yuqori darajali dasturlash tillari?
4. Yuqori darajali dasturlash tillarining bosqichlari?
5. Python dasturlash tilining imkoniyatlari?
6. Python dasturlash tilining boshqa dasturlash tillari bilan bog’liqligi?
7. Python dasturlash tilining keng tarqalish sabablari?
8. Python dasturlash tilining amaliy, web va matematik paket dasturlarini yaratish imkoniyati?
9. Python dasturlash tilining sohalarda qo‘llanilishi?

1.3-§. Python dasturlash tilining IDLE rejimida interaktiv dastur yaratish. Python dasturlash tilida faylli dastur yaratish.

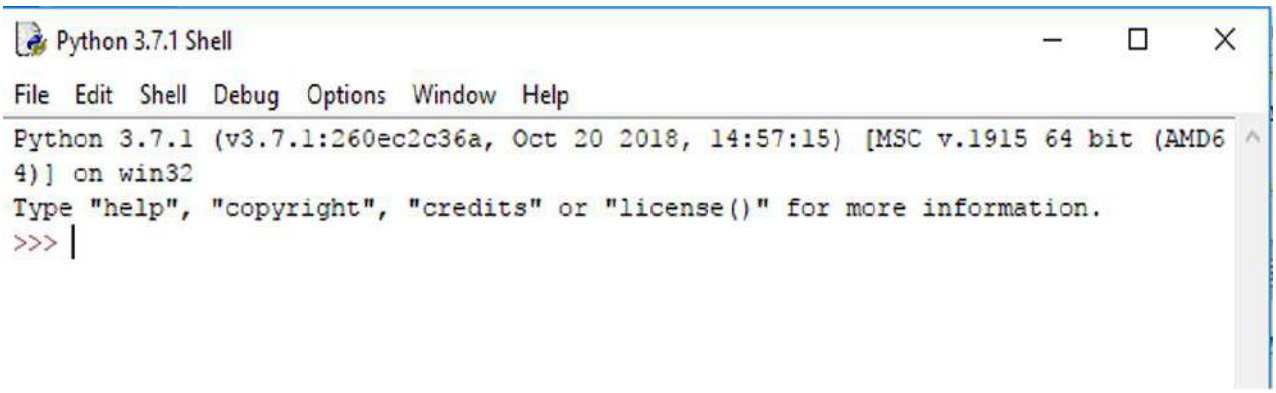
Reja:

1. *Python dasturlash tilining IDLE rejimida interaktiv dastur yaratish va o‘zgaruvchi turlaridan foydalanish.*
2. *Python dasturlash tilida faylli dastur yaratish.*

Tayanch so‘zlar: *IDLE rejimi, interaktiv, faylli dastur, ENTER, restart, int, float, str.*

Python dasturlash tili shaxsiy kompyuterga o‘rnatilgandan so‘ng IDLE rejimi yuklanadi. IDLE rejimi bu interaktiv rejim hisoblanadi, interaktiv rejimning qulayligi qisqa jarayonlarni dasturlashga juda qo‘l keladi. Bunda boshqa dasturlash tillari kabi dasturning umumiy tuzilishini to‘liq keltirish shart emas. O‘zgaruvchilar tabaqalashtirilgan holda e‘lon qilish shartlari keltirilmaydi, xotira bo‘yicha muammolar e‘tiborga olinmaydi.

Python dasturlash tili IDLE rejimi yuklangandan so‘ng quyidagi oyna hosil bo‘ladi.



1.3.1-rasm. Python dasturlash tili IDLE rejimining asosiy oynasi.

Python dasturlash tili IDLE rejimida birinchi dastur doimiy an’anaga muvofiq HELLO WORLD so‘zini ekranga chiqarishni qarab o‘tamiz. Satrli ma’lumotlar bittalik qo‘shtirnoq ichiga olib yoziladi.

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello, World');
      Hello, World
>>>
```

Bunda >>> belgidan so‘ng sonlarni kiritish va hisoblash jarayonlarini to‘g‘ridan to‘g‘ri amalga oshirish mumkin. Har bir son yoki amal kiritilgandan so‘ng ENTER tugmasi bosiladi va natijaga ega bo‘linadi.

Python tilida interaktiv rejim quyidagicha ishlatiladi.

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2.5
2.5
>>> 3.2+25
28.2
>>> 255656565956565+595659555656565
851316121613130
>>>
```

Bu jarayon bizga kalkulyator vazifasini ham bajaradi, kalkulyatordagi xotira muammosi bu yerda mavjud emas. Python dasturlash tili IDLE rejimida o‘zgaruvchilar bilan ishlash quyidagicha amalga oshiriladi.

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> a=52
>>> b=2.5
>>> a+b
54.5
>>> a*b+a/b
150.8
>>> a='maktab'
>>> a
'maktab'
>>> 2*a
'maktabmaktab'
>>> 3*a
'maktabmaktabmaktab'
>>>
```

Yuqoridagi dasturga e’tibor bersak, bunda o‘zgaruvchilar e’lon qilinmaydi interpretator qiymatga qarab turlarni aniqlaydi. Python dasturlash tilida buyruqlar boshqa dasturlash tillari kabi ; bilan tugatilish shart emas. Satrli ma’lumotlarga ko‘paytirish amali juda keng imkoniyatda ishlatilishi mumkin. Ixtiyoriy sonni satr ko‘rinishga satrni esa son ko‘rinishida foydalanish mumkin.

Int(satr) – bu funksiya satrni butun songa aylantiradi

Float(satr) – bu funksiya satrni haqiqiy songa aylantiradi

Str(son) – bu funksiya sonni satrga aylantiradi

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> a='123'
>>> int(a)+2
125
>>> a=123
>>> b=str(a)
>>> 2*b
'123123'
>>>
```

Haqiqiy sonlar faqat nuqta bilan yoziladi, vergul esa sonlarni bir biridan ajratish ucun xizmat qiladi. Python dasturlash tili IDLE rejimining qulayligi dasturlashni o‘rganayotganda yoki masala kodining ma’lum bir qismini sinab ko‘rish vaqtida juda qulaydir. Agar boshqa kompilyatsiya qilinadigan dasturlash tilda ishlasangiz, avval kodni asl dasturlash tilida yozishingiz, keyin kompilyatsiya qilishingiz kerak bo‘lgan faylni ishga tushirishingiz kerak bo‘ladi.

Python dasturlash tilida sonlarni uch xil turlari aniqlangan:

- Butun son

– Haqiqiy

- Kompleks

Python dasturlash tilida butun va haqiqiy sonlardan foydalanish tajribalarini yuqoridagi misollarda keltirib o‘tdik, complex sonlardan python dasturlash tilida quyidagicha foydalanamiz.

Complex(son) – bu funksiya sonni komleks songa aylantiradi

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> a=5
>>> b=complex(a)
>>> b
(5+0j)
>>> c=5+2j
>>> a+c
(10+2j)
>>>
```

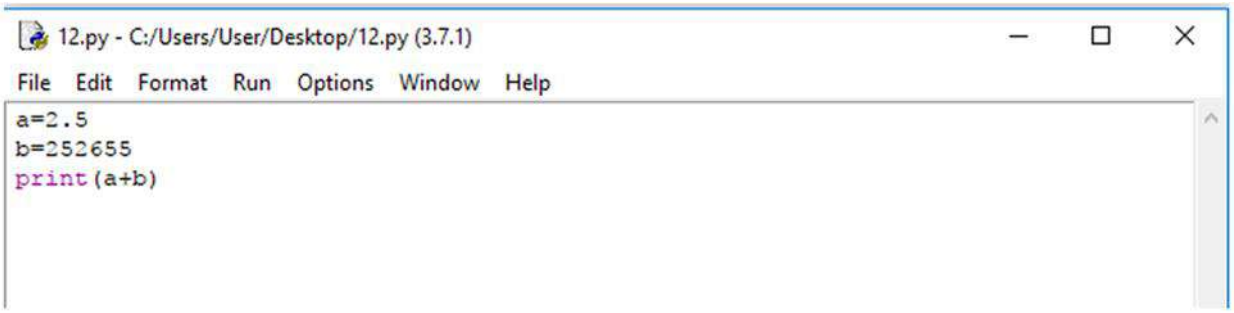
Python dasturlash tilining boshqa dasturlash tillaridan ustunligi kompleks sonlar ustida to‘g‘ridan to‘gri amal bajarish imkoniyati mavjudligi.

Python dasturlash tilida faylli dastur yaratish.

Python dasturlash tilida ko‘pgina hollarda, dasturchi masala kodini faylga saqlaydi va natijani fayl kod orqali amalga oshiradi. Bu jarayon boshqa dasturlash tillari kabi alohida fayl yaratish orqali datur tuziladi va yaratilgan dastur RUN tugmasi orqali ishga tushiriladi. Bu jarayon skript yozish deb nomlanadi.

Python dasturlash tilida yaratilgan fayl .py kengaytmaga ega bo‘ladi. Python dasturlash tilida skript yozish uchun IDLE interaktiv rejimida File → New File (yoki <Ctrl>+ N tugmachalarini bosing) ni tanlash orqali amalga oshiriladi. Yaratilgan faylga dastur tuzib RUN menyusi tarkibidan Run Module F5 komandasi tanlanadi, natijada dastur

natijasi interaktiv rejim oynasida aks etadi. Quyidagi dasturga e‘tibor bering.



Dastur natijasi quyidagicha bo‘ladi.

```
==============  RESTART:  C:/Users/User/Desktop/12.py
============
252657.5
>>>
```

Skript dastur yaratish jarayonida ma’lumotlarni kiritish(o‘qish) va chiqarish(yozish) operatorlari ishlatiladi. Bunda kiritish operatori ma’lumotlarni faqat satr ko‘rinishida qabul qiladi. Sonli ma’lumotlarni qayta ishlash uchun yuqoridagi int(); yoki float();funksiyasi yordamida amalga oshiriladi. Kiritish operatori o‘zgaruvchiga birlashtiriladi, kiritish operatorining umumiy ko‘rinishi quyidagicha.

o‘zgaruvchi = input() yoki o‘zgaruvchi = input(‘izoh’)

Kiritish operatorining umumiy ko‘rinishi quyidagicha.

Print(‘izoh’, o‘zgaruvchi)

Kiritish va chiqarish operatorlarini yozilishi quyidagi dastur tarkibida keltirilgan.

Misol. Ikkita a haqiqiy va b butun son berilgan bu sonlarni ko‘paytmasini ekranga chiqaring.

Yuqoridagi dasturda a=input('a='); izohni chiqargan holda a ga qiymat qabul qiladi, b=input() esa izohsiz b ga qiymat qabul qiladi. Input() funksiyasi tarkibida bir vaqtni o‘zida izohli va izohsiz kiritishni ishlatish mumkin. Kiritish operatori klaviaturadan kiritilgan ma’lumotlarni o‘qiydi va o‘zgaruvchan nomga yozadi. Chiqarish operatori esa o‘zgaruvchidagi ma’lumotni ekranga chop etadi.

```
a=input('a=');
b=input()
print('a+b=',float(a)*int(b));
Dastur natijasi
========== RESTART: C:/Users/User/Desktop/12.py
================
a=78545855.254454
5452154545
a*b= 4.282441417164835e+17
>>>
```

Nazariy savollar

1. Python dasturlash tilining IDLE rejimini ishga tushirish?

2. Python dasturlash tilining IDLE rejimida interaktiv dastur tuzish bosqichlari?

3. Int(), satr (), float() va complex() funksiyalarining vazifalari?

4.Komleks sonlardan foydalanish ular ustida amallar?

5.Python faylining kengaytmasi?

6.Pythonda faylli dastur tuzish bosqichlari?

7.Kiritish va chiqarish operatorlarining ko‘rinishlari va vazifalari?

Mustaqil ishlash uchu topshiriqlar

1. Kvadratning tomoni a berilgan. Uning perimetri P=4·a va yuzasi aniqlansin. $S=a^2$.

2. To‘g'ri to‘rtburchakning tomonlari a va b berilgan. Uning yuzasi S=a·b; va P=2·(a+b) perimetri aniqlansin.

3. Aylananing diametri d berilgan. Uning uzunligi aniqlansin L=pi·d.

4. Kubning yon tomoni a berilgan. Uning hajmini V=a3 va to‘la sirti $S=6\cdot a^2$ aniqlansin.

5. Paralelepepidning tomonlari a, b, c berilgan. Uning hajmi V=a·b·c va to‘la sirti S=2·(a·b+b·c+a·c) aniqlansin.

6. Doiraning radiusi R berilgan. Uning uzunligi L va yuzasi S aniqlansin. L=2·pi·R, $S=pi\cdot R^2$.

7. Ikkita son a va b berilgan. Ularning o‘rta arifmetigi aniqlansin. (a+b)/2 va o‘rta geometrigi aniqlansin.

8. Nolga teng bo‘lmagan ikkita son berilgan. Ularning yig‘indisi, ko‘paytmasi va har birining kvadrati aniqlansin.

9. Nolga teng bo‘lmagan ikkita son berilgan. Ularning yig‘indisi, ko‘paytmasi va har birining moduli aniqlansin.

10. To‘g'ri uchburchakning katetlari a va b berilgan. Uning gipotenuzasi c va perimetri P aniqlansin. .

11. Umumiy markazga bo‘gan ikkita aylana radiusi berilgan. R1 va R2 (R1 > R2). Ularning yuzalari S1 va S2, ularning ayirmasi S3 aniqlansin. $S1=pi\cdot(R1)^2$, $S2=pi\cdot(R2)^2$, S3=S1–S2.

12. Aylananing uzunligi L berilgan. Uning radiusi R va yuzasi S aniqlansin. $L=2\cdot pi\cdot R$, $S=pi\cdot R^2$.

13. Aylananing yuzasi S berilgan. Uning diametri D va uzunligi L aniqlansin. $L=2\cdot pi\cdot R$, $S=pi\cdot R^2$.

14. Sonlar o„qida A, B, C nuqtalar berilgan. AC va BC kesmalarning uzunligini va kesmalar uzunligining yig'indisini topuvchi algoritm tuzilsin.

15. To„g'ri to„rtburchakning qarama-qarshi uchlari koordinatlari berilgan. Uning tomonlari koordinata o„qiga parallel. To„g'ri to„rtburchakning perimetri va yuzasi aniqlansin.

16. Tekislikda berilgan ikki nuqta (x1, y1) va (x2, y2) orasidagi masofa topilsin.

17. Uchburchakning uchta tomoni uchlari koordinatlari berilgan (x1, y1), (x2, y2), (x3, y3). Ikki nuqta orasidagi masofa topilsin.

18. A, B va C sonlari berilgan. A ni qiymati B ga, B ni qiymati C ga, C ni qiymati A ga almashtirilsin. A, B va C ning yangi qiymatlari ekranga chiqarilsin.

1.4-§. Python dasturlash tilining standart kutubxonalari va tashqi kutubxonasini o‘rnatish.

Reja:

1. *Kutubxonani cmd orqali o‘rnatish.*
2. *Tashqi kutubxonalarni ishlatish*

Tayanch so‘zlar: *pip, cmd, package, matplotlib, numpy, wordcloud, pandas.*

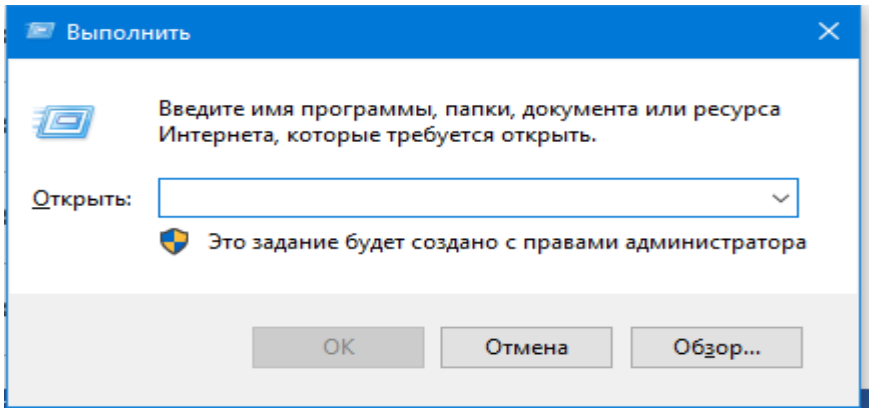
Python bilan birga o‘rnatiluvchi, standart kutubxona va undagi ba’zi foydali modullar bilan tanishganmiz. Ushbu bo‘limda esa tashqi kutubxona bilan tanishamiz. Bu kutubxonalar yillar davomida turli dasturchilar tarafidan yaratilib, yangilanib kelinadi. Bunday kutubxonalarda boshqa dasturchilar o‘zlari yaratgan turli paketlarni (package) boshqalar bilan ulashadi.

Paket (package) —modullar yig‘indisi.

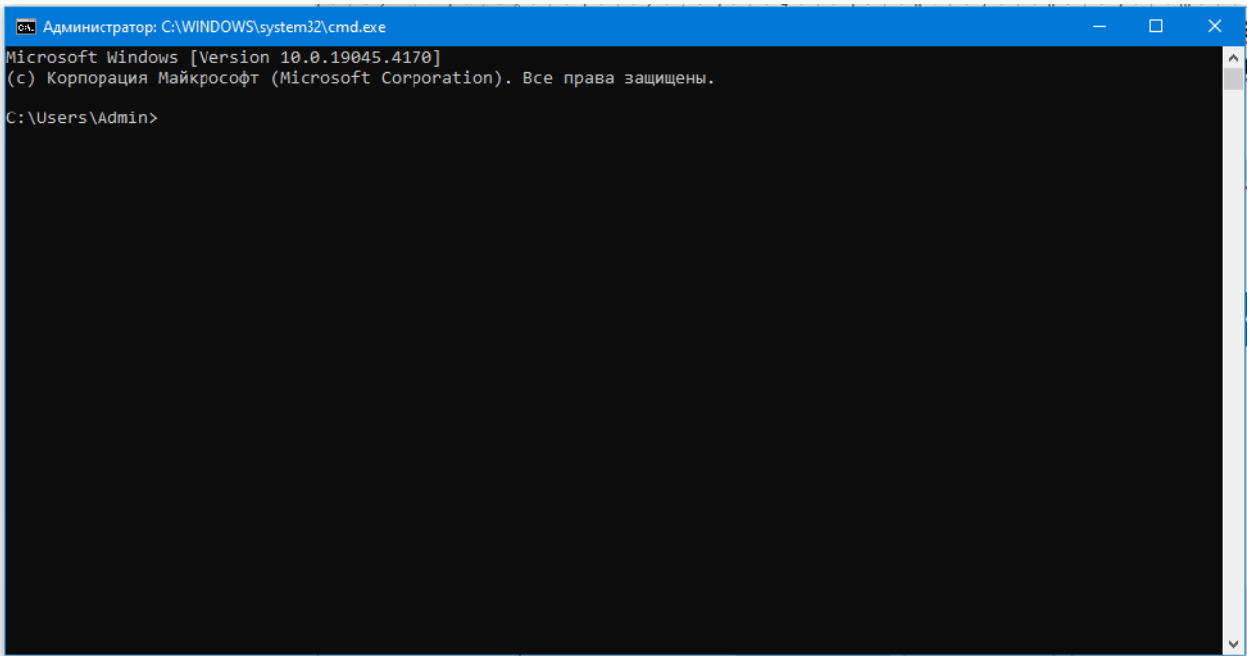
Tashqi kutubxonalar va ular ichidagi paketlar shunchalik ko‘pki, deyarli istalgan vazifa yoki xizmat uchun katta ehtimollik bilan kerakli dasturlar allaqachon bir nechtadan yaratilgan. Bugungi kunda Python uchun eng katta tashqi kutubxonalardan biri bu PyPi.org sahifasi.

Pip orqali python kutubxonalarini cmd orqali o‘rnatish

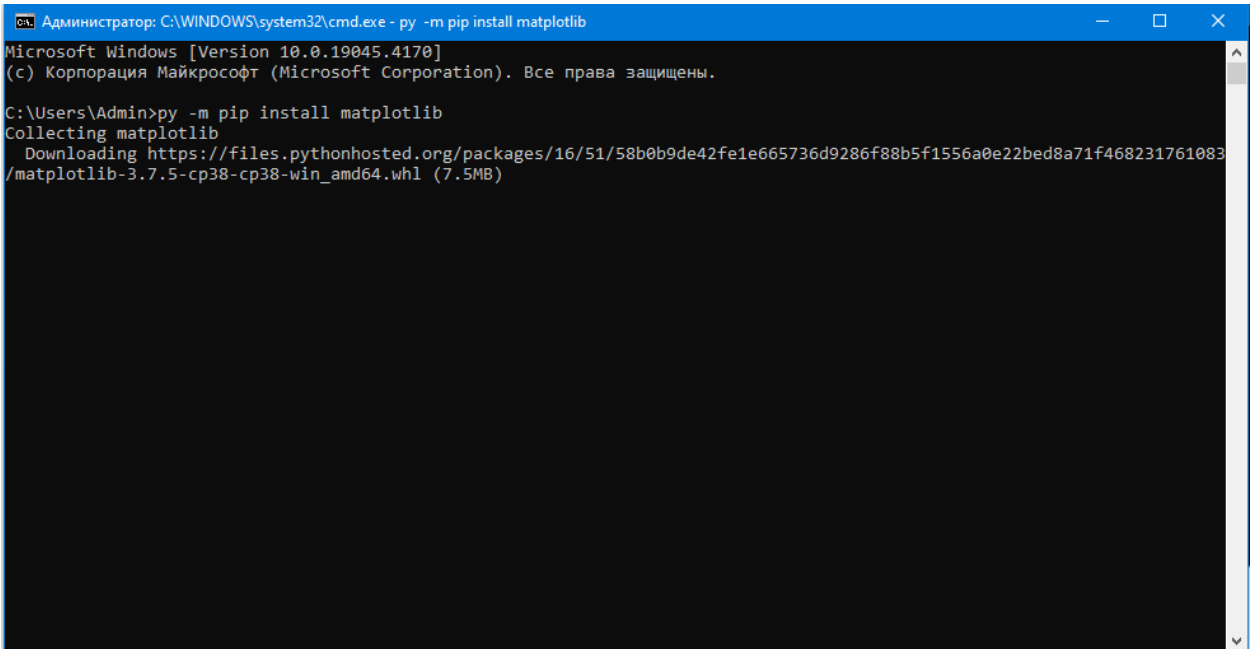
Win+R tugmachalarini bosamiz va quyidagi oyna hosil bo‘ladi.



Hosil bo‘lgan oynaga cmd buyrug‘ini yozib OK tugmasini bosamiz va natijada quyidagi oyna hosil bo‘ladi.



py -m pip install matplotlib Shu tariqa boshqa kutubxonalarni ham o‘rnatib olamiz.



- Veb-ishlab chiqish uchun Python kutubxonalari
  - HTTPX
  - Celery
  - Scrapy
  - Dash
- Mashina o‘rganish uchun Python kutubxonalari
  - NumPy
  - Pandas
  - TensorFlow
  - LightGBM
- Ma’lumotlarni formatlash va tozalash uchun Python kutubxonalari
  - Dora
  - Datacleaner
  - Tabulate
  - Scrubadub
- Ma’lumotlarni vizualizatsiya qilish uchun Python kutubxonalari
  - Matplotlib
  - Altair
  - Bokeh
  - Leather
- Telegram botlarini yaratish uchun Python kutubxonalari
  - AIogramma
  - Python-telegram-bot
  - Telebot

2023 yil boshiga kelib, Python eng mashhur dasturlash tillaridan biri bo‘lib qolmoqda. Sintaksisning soddaligi va qulayligidan tashqari, u ko‘plab plagin kutubxonalariga ham ega. Python-da ularning 137 000 dan ortig‘i bor va bu raqam deyarli har kuni o‘sib bormoqda.

Tashqi paketlarni o‘rnatish uchun Pythonda maxsus pip paket menejeri mavjud. pip odatda Python bilan birga o‘rnatiladi, lekin turli sabablarga ko‘ra kompyuteringizda pip o‘rnatilmagan bo‘lsa, uni quyidagi sahifadan yuklab olishnigiz mumkin: https://pypi.org/project/pip/

Paket menejer yordamida tashqi paketlarni o‘rnatish juda oson, buning uchun Windows terminalda (cmd) (yoki Spyder konsolida, yoki PyCharm konsolida va hokazo)pip install paket_nomi komandasidan foydalanasiz.

Paket nomi qanday yozilishini paketning rasmiy sahifasidan ko‘rib olishingiz mumkin.

Googlening tarjimonlik xizmatiga murojat qilib, istalgan matnni turli tillarga tarjima qilishimiz mumkin. Moduldan foydalanish uchun avvalo googletrans modulidan Translator klassini import qilamiz va bu klassdan yangi obyekt yaratamiz (tarjimon). Bevosita tarjimonlik xizmatiga murojat qilish uchun tarjimon obyekti ichidagi .translate() metodiga murojat qilamiz va parametr sifatida tarjima qilish kerak bo‘lgan matnni uzatamiz.

```
from googletrans import Translator

tarjimon = Translator() # Translator bu maxsus klass (tarjimon esa obyekt)

matn_uz = "Python - dunyodagi eng mashxur dasturlash tili"

tarjima = tarjimon.translate(matn_uz)

print(tarjima.text)
```

Natija: Python is the most popular programming language in the world

DIQQAT! Agar yuqoridagi kod xato bersa (TKK token error), googletrans modulini o‘chirib tashlab, huddi shu modulning yangi versiyasini o‘rnating:

```
pip uninstall googletrans # modulni o‘chirish

pip install googletrans==3.1.0a0 # yangi verisyani o‘rnatish
```

Agar boshqa tillarga tarjima qilish kerak bo‘lsa, .translate() metodiga matnga qo‘shimcha ravishda test parametrini ham uzatamiz va bu parametrga tarjima qilinishi kerak bo‘lgan tilning qisqartmasini beramiz. Tarjima uchun mavzuda tillarni quyidagi manzilga ko‘rishingiz mumkin:

https://sites.google.com/site/opti365/translate_codes

Masalan, rus tiliga tarjima qilish uchun test='ru' deb yozamiz.

```
tarjima_ru = tarjimon.translate(matn_uz, test='ru')

print(tarjima_ru.text)
```

```
Natija: Python - самый популярный язык программирования в мире
```

Ingliz tilidan boshqa tillarga tarjima ham shunday:

```
matn_in = "Tashkent is the capital of Uzbekistan"
tarjima_uz = tarjimon.translate(matn_en, test='uz')
 print(tarjima_uz.text)
```

```
Natija: Toshkent - O‘zbekistonning poytaxti
```

Odatda, Google asl matnning tilini o‘zi aniqlaydi. Lekin matn tilini ham alohida ko‘rsatmoqchi bo‘lsangiz, src parametridan foydalaning:

```
tarjima_uz = tarjimon.translate(matn_in, src='uz', test='uz')
```

REQUESTS

Bu paket yordamida Pythonda veb sahifalarga murojat qilishimiz (so‘rov yuborishimiz) va ulardan qaytgan ma’lumotlar ustida turli amallar bajarishimiz mumkin. Misol uchun quyida requests yordamida kun.uz sahifasini to‘liqligicha toritb olamiz:

```
import requests
from print import print
manzil= "https://kun.uz/news/main"
r = requests.get(manzil)
print(r.text)
```

Ko‘pincha requests paketidan APIlar bilan ishlashda foydalaniladi. API bu ma’lum bir veb hizmatga so‘rov yuborish orqali undan foydalanish. Misol uchun yandex tarjimonga yoki google xaritalari xizmatiga requests paketi yordamida API so‘rov yuborish va o‘zimizga kerakli ma’lumotlarni olishimiz mumkin.

Internetda restcountries.eu sahifasi mavjud. Bu sahifa orqali dunyodagi davlatlar haqida turli ma‘lumotlarni olishingiz mumkin. Sahifadan foydalanish qulay bo‘lishi uchun esa, sahifa yaratuvchilari bir nechta tayyor API lar e’lon qilishgan. Misol uchun O‘zbekiston haqida ma‘lumot olish uchun quyidagi manzilga so‘rov yuborasiz: https://restcountries.eu/rest/v2/name/Uzbekistan

API dan qaytgan natija JSON (lug‘at) ko‘rinishda bo‘ladi va biz bu lug‘atdan o‘zimizga kerakli ma‘lumotni sug‘urib olishimiz mumkin. Misol uchun quyidagi kodimiz APIga yuborilgan davlatning poytaxtini ko‘rsatadi:

```
country = "Uzbekistan"

url = f"https://restcountries.eu/rest/v2/name/{country}"

r = requests.get(url)

r_json = r.json()[0]

print(r_json['capital'])
```

Natija: Tashkent

BEAUTIFULSOUP4

BeautifulSoup juda kuchli modullardan biri bo‘lib, bu modul yordamida turli veb sahifalardan istalgan ma’lumotlarni yig‘ishtirib (scarpping) olish mumkin. Biror kishining instagram sahifasidagi barcha rasmlar deysizmi, Facebook guruhidagi barcha postlar va izohlar deysizmi, oldi-sotdi bozoridagi e'lonlar deysizmi, marhamat, bs4 moduli yordamida buni bemalol avtomatlashtirish mumkin.

Odatda bs4 moduli requests moduli bilan hamkorlikda ishlaydi. Keling, sodda misol ko‘ramiz. Avvalgi bo‘limda, requests yordamida kun.uz sahifasining html kodini olgan edik. Endi esa bs4 yordamida html sahifadan oxirgi yangiliklarning mavzusini ajratib olamiz.

```
import requests
from bs4 import BeautifulSoup
url = 'http://mignews.com/mobile'
page = requests.get(url)
print(page.status_code)
filteredNews = []
allNews = []
soup = BeautifulSoup(page.text, "html.parser")
print(soup)
allNews = soup.findAll('a’, class_='lenta’)
for data in allNews:
    if data.find('span', class_='time2 time3') is not None:
        filteredNews.append(data.text)
    for data in filteredNews:
        print(data)
```

Natijasi: Web sahifaning html kodlarini olib beradi

wordcloud va matplotlib

pip install wordcloud

pip install matplotlib

Wordcloud moduli yordamida katta matnlarda eng ko‘p uchraydigan so‘zlarni chiroyli qilib, so‘zlar buluti chiqarish mumkin. 2020-yil yakunida, sariqdev sahifasida chop etilgan mashxur blogerlarning siluetlari ham aynan shu modul yordamida qilingan.

wordcloud moduli mahsuli

Quyida kun.uz sahifasidan olingan yangiliklar uchun so‘zlar bulutini yaratishni ko‘ramiz.

```
import requests
from bs4 import BeautifulSoup
from wordcloud import WordCloud
import matplotlib.pyplot as plt


sahifa = "https://kun.uz/news/main"
r = requests.get(sahifa)

soup = BeautifulSoup(r.text, 'html.parser')
news = soup.find_all(class_="news-title")
matn=""
for n in news:
    matn += n.text

# kerakmas so'zlar
stopwords=
["учун","бўйича","лекин","билан","ва","бор","ҳам","хил","йил"]
# bulutni yaratamiz
wordcloud = WordCloud(width = 1000, height = 1000,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 20).generate(matn)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```

Nazariy savollar.

1. Python dasturlash tilida kutubxona qanday chaqiriladi?
2. Matplotlib kutubxonasi haqida tushuncha bering.
3. Tashqi kutubxona qanday o‘rnatiladi?
4. Win+R tugmasini bosganimizda qanday oyna hosil bo‘ladi.
5. BeautifulSoup moduli haqida ma’lumot bering.

Mustaqil ishlash uchun topshiriqlar.

1. Python dasturida matematik kutubxonasidan foydalang.
2. Python dasturida turtle kutubxonasi orqali kvadtar chizing.
3. Random kutubxonasidan foydalanib 1 dan 50 gacha bo‘lgan sonlar ichidan tasodifiy 10 sonni aniqlash dasturini tuzing.
4. Berilgan a butun sonni kvadrat ildizini hisoblash dasturini tuzing.
5. Bugungi sanadan boshlab 2 hafta farq bilan 10 ta sanani konsolga chiqaring.
6. Tug'ilgan kuningizdan bugungi sanagacha qancha yil, oy, kun o‘tganini qaytaruvchi funksiya yozing.
7. Foydalanuvchidan telefon raqamini kiritishni so‘rang. Kiritlgan qiymatni andoza yordamida tekshiring.
8. Berilgan matndan veb sahifa manzilini ajratib olyuvchi funksiya yozing.
9. Kvadratik funksiyaning ildizlarini toppish uchun Python dasturini yozing.
10. Floatni kasrga aylantirish uchun Python dasturini yozing.

1.5-§. Python dasturlash tili tarkibidagi matematik funksiyalar va ifodalar.

Reja:

1. Python tilida ifodalar;

2. Python dasturlash tilida matematik funksiyalar.

Tayanch so‘zlar: *ifoda, import, math, standart funksiya, min, max,sum.*

Barcha dasturlash tillari kabi Python dasturlash tilida ham matematik ifodalar ma’lum bir standartlar asosida yoziladi. Ifodalar tarkibidagi matematik funksiyalar Python tilida standart fuksiyalar yordamida yoziladi, agar ifoda tarkibidagi funksiya standart funksiya tarkibida bo‘lmasa, alohida funksiya yaratib olish kerak.

Ifoda-sonlar, harflarni arifmetik amallar va qavslar bilan birlashtirilgan yozuvga aytiladi.

Python dasturlash tilidagi ifodalar tarkibidagi amallarni bajarilishi matematikadagi amallarni bajarilish tartibiga mos keladi. Python tilida arifmetik amallarni yozilishi yuqoridagi mavzuga asosan yoziladi. Ifodalar tarkibidagi nomalumlar faqatgina lotin alifbosida yozilishi kerak. Ifoda tarkibida kasr sonning surati yoki maxrajida ikki va undan ortiq hadlar bo‘lsa, python tilida ular albatta qavsga olinishi kerak.

Python dasturlash tilida matematik funksiyalar

Python dasturlash tili tarkibida mavjud bo‘lgan matematik funksiyalar **standart funksiyalar** deb ataladi.

Ifodalar tarkibidagi funksiyalarni Python dasturlash tilida ifodalash uchun satandart funksiyalardan foydalaniladi. Funksiyalarni python dasturlash tilida ifodalash uchun ularni argumentlarini albatta qavsga olib yozish kerak.

Python dasturlash tilida matematik funksiyalardan foydalanish uchun albatta python tili tarkibidagi matematik funksiyalar kutubxonasiga murojat qilish kerak. Matematik funksiyalar kutubxonasiga murojat qilish quyidagicha.

from math import*

Python dasturlash tili tarkibidagi matematik funksiyalar yozilishi quyidagi ro‘yxat asosida amalga oshiriladi.

№	Python dasturlash tilida ifodalanishi	Matematik ifodalanishi
1	*trunc(x)*	*Sonning butun qismi trunc(5.8)=5, trunc(-5.8)=-5*
2	*sqrt(x)*	*x ning kvadrat ildiz*
3	*log(x), log2(x), log10(x)*	$\ln x$, $\log_2 x$, $\log_{10} x$
4	*log(x,a)*	$\log_a x$
5	*sin(x), cos(x), tan(x)*	*Sinx, cosx, tgx trigonametrik funksiyalar*
6	*asin(x), acos(x), atan(x)*	*Arcsinx, arccosx, arctgx teskari trigonametrik funksiyalar*
7	*atan2(x,y)*	
8	*degrees(x)*	*Radiandan gradusga o‘tkazish funksiyasi*
9	*radians(x)*	*Gradusdan radianga o‘tkazish funksiyasi*
10	*sinh(x), cosh(x), tanh(x)*	*Giperbolik Sinx, cosx ,tgx trigonametrik funksiyalar*
11	*asinh(x), acosh(x), atanh(x)*	*Giperbolik Arcsinx, arccosx, arctgx teskari trigonametrik funksiyalar*

12	*hypot(x,y)*	*Katetlari x va y bo'lgan to'g'ri burchakli uchburchakning getatinuzasini topish*
13	*factorial(x)*	*X faktorialni hisoblash funksiyasi*
14	*gamma*	*x ning gamma funksiyasi*
15	*pi*	*π soni π=3.1415…*
16	*e*	*Eksponentsial funksiaya e=2.71…*

Matematik funksiyalarning bajarilish jarayoni, matematikada qanday bo‘lsa python dasturlash tilida ham xuddi shunday amalga oshiriladi.

Matematik funksiyalarni ishlash jarayoni tushunarli bo‘lishi uchun, ularni interaktiv rejimda sinab ko‘ramiz. Chunki interaktiv rejim bir vaqtning uzida natija qaytaradi.

Misol. Matematik funksiyalarni bajarilishi

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> from math import*
>>> x=12.7
>>> trunc(x)
12
>>> trunc(12.2)
12
>>> trunc(-12.7)
-12
>>> trunc(-12.2)
-12
>>> sqrt(81)
9.0
>>>log(e),log2(8),log10(x)
(1.0, 3.0, 1.1038037209559568)
>>> log(81,3)
4.0
 >>> cos(pi)
-1.0
>>> atan(180)
1.565240828394204
>>> radians(180), degrees(pi/3)
(3.141592653589793, 59.99999999999999)
```

Qo‘shimcha funksiyalar

Python dasturlash tilida standart kutubxona tarkibida ketma ketliklar ustida bir nechta maximum, minimum, summa kabi funksiyalar aniqlangan. Boshqa dasturlash tillarida bu funksiyalar alohida algoritmlar yordamida tuzib olinadi, python dasturlash tilida esa bu funksiyalar tayyor holda saqlanadi. Qo‘shimcha funksiyalar quyidagi jadval ko‘rinishida amalga oshiriladi.

№	Python dasturlash tilida ifodalanishi	Matematik ifodalanishi
1	*max(a,b,...)*	*Sonlar yoki kortej ichidan eng kattasini topish. max(2,-8)=2*
2	*min(a,b,...)*	*Sonlar yoki kortej ichidan eng kichigini topish. min(2,-8)=-8*
3	*sum(a,b,...)*	*Sonlar yoki kortej yig'indisini topish. sum(2,-8)=-6*
4	*sorted(a,b,...)*	*Sonlarni tartiblash. sorted(3,12,-9)=(-9,3,12)*

Qo‘shimcha funksiyalarni ishlash jarayoni tushunarli bo‘lishi uchun, ularni interaktiv rejimda sinab ko‘ramiz. Chunki interaktiv rejim bir vaqtning uzida natija qaytaradi.

Misol. Qo‘shimcha funksiyalarni bajarilishi

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> max(12,5,-8,7)
12
>>> min(12,5,-8,7)
-8
>>> a=(1,2,3,0,-12,123)
>>> a
(1, 2, 3, 0, -12, 123)
>>> max(a), min(a), sum(a)
(123, -12, 117)
>>> sorted(a)
[-12, 0, 1, 2, 3, 123]
>>>
```

Misol. Quyidagi ifodalarni python tilida ifodalash.

Matematik ifodasi

$$y = (x + \sin x)^3 - \cos^2 x + \frac{1 + \log_a x}{\sqrt{t - x^2}}$$

Python tilida ifodalanishi

y=pow((x+sin(x)),3)+sqr(cos(x))+(1+log(x,a))/(sqrt(t-x**2));

Misol. Quyidagi ifodalarni Python tilida ifodalash.

Matematik ifodasi

$$y = |x - 2| + \sin x - \left|\frac{4}{\sqrt{t - x^2}}\right|$$

Python tilida ifodalanishi

y=abs(x-2)+sin(x)-abs(4/sqrt(t-x**2));

Nazorat savollari

1.Ifoda deb nimaga aytiladi?

2.Ifodalarni python tilida yozilish tartibi?

3.Standart funksiya deb nimaga aytiladi?

4.Standart funksiyalarni python tilida yozilish tartibi?

5.Matematik kutubxonaga murojat qilish tartibini tushuntirib bering?

6.Matematik funksiyalarning har birini yozib ma’nosini tushuntirib bering?

7.Qo‘shimcha(min, max, sum) funksiyalarning har birini yozib ma’nosini tushuntirib bering?

Mustaqil ishlash uchun topshiriqlar

Quyidagi ifodalarni python dasturlash tilida yozing

$$y = \frac{1 + x^4}{sinx + |x|} - tgx$$

$$y = \frac{ax^2 - bx + c}{|x|} - \left|\frac{x - t}{\sqrt{x + a}}\right|$$

$$y = \frac{2}{|ax^2 - bx + c|} - \left|\frac{x - t}{\sqrt{x + a}}\right|$$

$$y = \frac{\sqrt{x + a}}{sinx + |x|} - 1 + x^4$$

$$y = \frac{\sqrt{x + a}}{sinx + |x|} - ax^2 - bx + c$$

$$y = ax^2 - bx + c - \left|\frac{log_a x}{\sqrt{x + a}}\right|$$

$$y = \frac{\sqrt{x + a}}{sinx + |x|} - \left|\frac{log_a x}{\sqrt{x + a}}\right|$$

$$y = \pi - 3x - \frac{\sqrt{x + a}}{sinx + |x|}$$

$$y = \pi - 3x - \left|\frac{log_a x}{\sqrt{x + a}}\right|$$

$$y = \pi - ax^2 - bx + c - \frac{\sqrt{x + a}}{sinx + |x|}$$

II BOB. PYTHON DASTURLASH TILIDA CHIZIQLI, TARMOQLANUVCHI VA TAKRORLANUVCHI JARAYONLARNI DASTURLASH.

2.1-§. Dasturlash tili tarkibidagi o‘zgaruvchi va o‘zgarmaslarni e’lon qilish.

Reja:

1. *Python o‘zgaruvchilari, konstantalari.*
2. *Python-da o‘zgaruvchilar turlari va ularga qiymat berish usullari.*
3. *O‘zgaruvchining qiymatini o‘zgartirish*

Tayanch so‘zlar: *int, float, str, son, o‘zgaruvchi, o‘zgarmas, constant.*

Dasturlash tilining asosi ma’lumotlar elementlari va ular saqlanadigan blokni o‘z ichiga oladi. Ularning har biri uchun maxsus nomlar berilgan va ularda maxsus funksiyalar bajarilishi mumkin. Python dasturlash tilida ular o‘zgaruvchilar, doimiylar va harflar deb ataladi. Ushbu mavzuda biz Python konstantalari va o‘zgaruvchilari, va ularni elon qilish, ularning turlari va misollari bilan ko‘rib chiqamiz.

Python o‘zgaruvchilari:

O‘zgaruvchi - ***bu*** *dastur ishlayotgan vaqtda ma’lumotlarni saqlash uchun nomlangan joy. Dasturlash tilida* ***o‘zgaruvchilar*** *har qanday turdagi ma’lumotlarning qiymatlarini saqlash uchun ishlatiladigan so‘zlardir.*

Oddiy so‘zlar bilan aytganda, siz o‘zgaruvchini yaratganingizda, u siz o‘rnatgan qiymat va turga qarab bir oz xotira maydonini egallaydi. Python tarjimoni RAMni o‘zgaruvchiga ma’lumotlar turiga qarab ajratadi. O‘zgaruvchilar qiymatlari dastur davomida istalgan vaqtda o‘zgartirilishi mumkin.

o‘zgaruvchi_nomi = ma’lumotlar qiymatlari

bu erda, o‘zgaruvchi_nomi = harflar, raqamlar va pastki chiziq birikmasi bo‘lishi mumkun.

O‘zgaruvchi nomini e'lon qilishda amal qilish kerak bo‘lgan qoidalar

1. O‘zgaruvchining nomi raqam bilan boshlanmaydi. Birinchi belgi sifatida alifbo yoki pastki chiziq belgisi ishlatilishi kerak.
2. O‘zgaruvchilar nomlari katta-kichik harflarga sezgir bo‘lib, alfanumerik harflar bilan bir qatorda pastki chiziq belgisini ham o‘z ichiga olishi mumkin.
3. O‘zgaruvchilar nomlarida zaxiralangan shartlar bo‘lishi mumkin emas.
4. Pythonda o‘zgaruvchilarni tayinlash uchun **‘=’** belgisidan keyin o‘zgaruvchining qiymatidan foydalaniladi.

Python-da o‘zgaruvchilar turlari va ularga qiymat berish usullari

Python dasturlash tilida buyruqlar boshqa dasturlash tillari kabi ; bilan tugatilish

shart emas. Satrli ma’lumotlarga ko‘paytirish amali juda keng imkoniyatda ishlatilishi mumkin. Ixtiyoriy sonni satr ko‘rinishga satrni esa son ko‘rinishida foydalanish mumkin.

Int(satr) – bu funksiya satrni butun songa aylantiradi

Float(satr) – bu funksiya satrni haqiqiy songa aylantiradi

Str(son) – bu funksiya sonni satrga aylantiradi

O‘zgaruvchiga ma’lumotlar elementlarini belgilashning bir necha xil usullari mavjud. Eng keng tarqalganlari quyida tavsiflanadi.

Satrga oid funksiyalar

Pythonda satr bilan ayrim amallarni bajarish uchun maxsus funksiyalar bor. Ularning soni ancha ko‘p, hammasini eslab qolish esa qiyin. Shuning uchun kerakli fuksiyani manbadan tanlab foydalangan ma’qul. Hozir esa shulardan ba’zilarini ko‘rib chiqamiz.

Metodlarni chaqirganga Pythondagi satrlar o‘zgarmaydigan ketma-ketliklar darajasiga kirishini inobatga olishimiz kerak. Bu degani hamma funksiyalar va metodlar faqat yangi satrni tuzishi mumkin.

S.rfind(str,[start],[end])- Satrdan satr ostini axtarish. Oxirgi kirish raqamini yoki 1 ni qaytaradi

S.index(str,[start],[end])- Satrdan satr ostini axtarish. Birinchi kirish raqamini qaytaradi yoki ValueError istisnosini chaqiradi

S.rindex(str,[start],[end])- Satrdan satr ostini axtarish. Oxirgi kirish raqamini qaytaradi yoki ValueError istisnosini chaqiradi.

S.isdigit() - Satrda raqamlar ishtirok etganligini tekshirish.

S.isalpha() - Satr faqat harflardan iboratligini tekshirish.

S.isalnum() - Satr harf yoki raqamlardan iboratligini tekshiradi.

S.islower() - Satr quyi registrdagi belgilardan iboratligini tekshiradi**.**

S.isspace()-Satrda ko‘rinmaydidan belgilar borligini tekshirish (probel, sahifani o‘tkazish belgisi(‘\p’), yangi satrga o‘tish(‘\n’), koretkani qaytarish(‘\r’), gorizontal tabulyatsiya(‘\t’) va vertikal tabulyatsiya)

S.istitle() - Satrda so‘zlar bosh harf bilan boshlanishini tekshirish.

S.starswith(str)- S satr str shablonidan boshlanishini tekshirish.

S.endswith(str)- S satr str shabloni bilan tugashini tekshirish.

S.join(ro‘yxat)- S ajratuvchiga ega ro‘yxatdan qatorni yig‘ish.

Ord(belgi)- Belgiga mos ASCII kodni qaytaradi.

Chr(son)- ASCII kodga mos belgini qaytaradi.

S.capitalize()-Satrning birinchi belgisi yuqori registrda qolganlarini quyi registrga o‘tkazadi.

S.center(width,[fill])- Chegaralari bo‘yicha fill (jimlik holatida probel) belgisi turuvchi markazlashtirilgan satrni qaytaradi.

S.expandtabs(tabsize)- Joriy ustungacha bir yoki bir qancha probellar bilan tabulyatsiyaning hamma belgilari almashtirilgan satr nusxasini qaytaradi. Agarda TabSize ko‘rsatilmagan bo‘lsa tabulyatsiya hajmi 8 probelga teng bo‘ladi.

S.lstrip([chars])- Satr boshidagi probel belgilarini olib tashlash.

S.rstrip([chars]) - Satr oxiridan probel belgilarini olib tashlash.

S.strip([chars]) - Satr boshidan va oxiridan probel belgilarini olib tashlash.

S.partition(shablon) - Birinchi shablon oldida turuvchi qismni keyin shablonni o‘zini va shablondan keyin turuvchi qismga ega kortejni qaytaradi. Agarda shablon topilmasa satrga ega bo‘lgan kortejni qaytaradi, avval ikki bo‘sh satr keyin satrni o‘zini.

S.rpartition(sep)- Oxirgi shablon oldida turuvchi qismni keyin shablonni o‘zini va shablondan keyin turuvchi qismni qaytaradi. Kortej qator o‘zidan va undan keyin ikkita bo‘sh qatordan iborat bo‘ladi.

S.swapcase()-Quyi registrdagi belgilarni yuqori registrga, yuqorilarni esa quyiga o‘tkazadi.

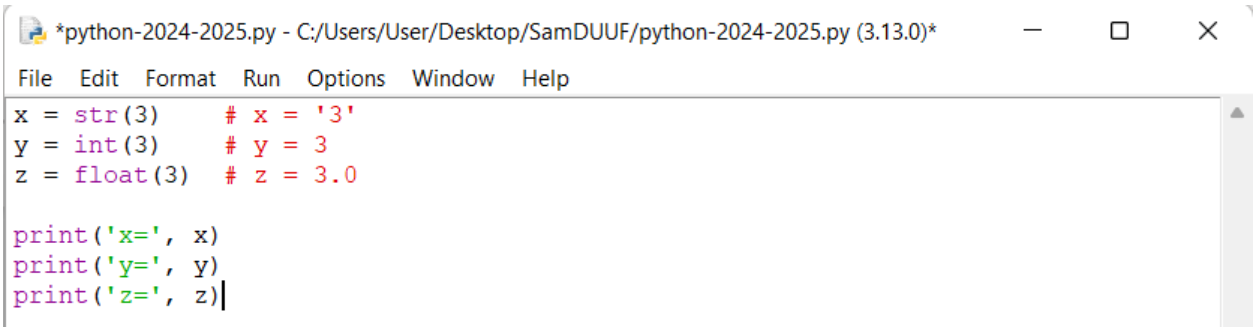
S.title() - Har bitta so‘zning birinchi harfini yuqori registrga qolganlarini esa quyi registrga o‘tkazadi.

S.zfill(width) - Qator uzunligini Widthdan kam qilmaydi agar kerak bo‘lsa birinchi belgilarni nollar bilan to‘ldiradi.

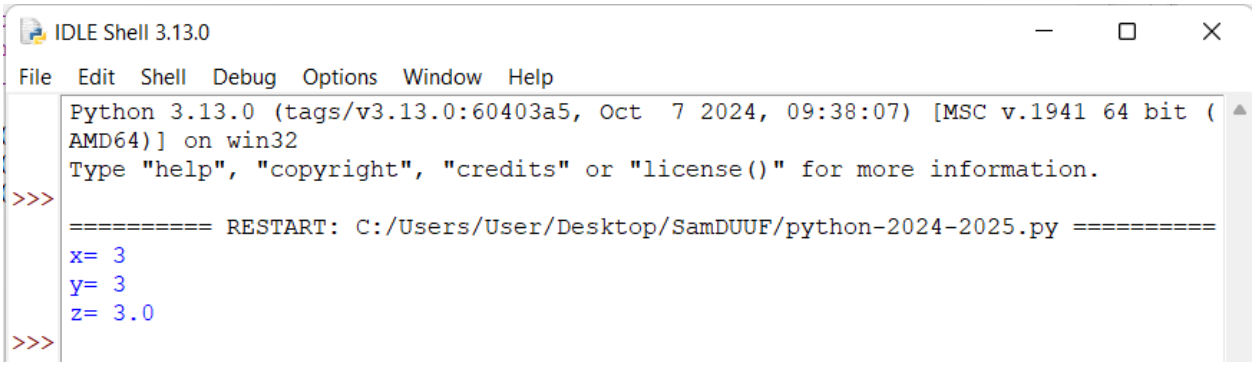
S.strip() funksiyasi satrning boshi va oxirida bo‘shliqlar bo‘lsa, olib tasdiqlaydi:

O‘zgaruvchilar turlarini aniqlash

Misol:



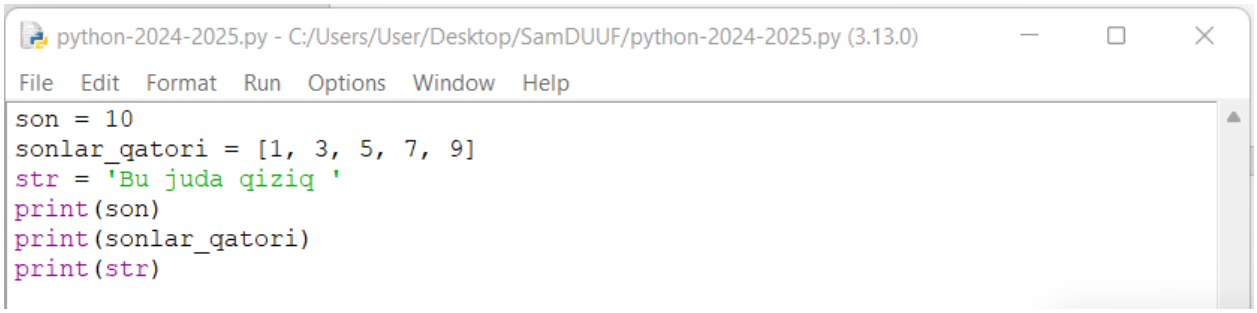
Natija



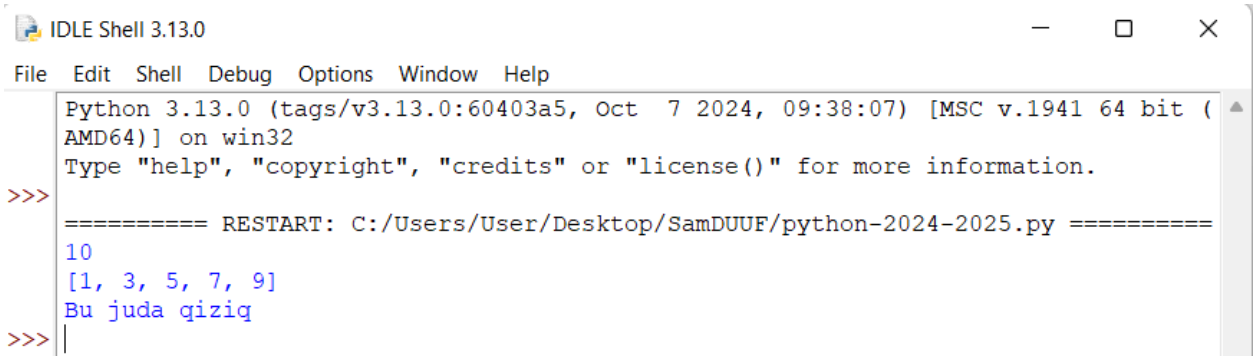
Oddiy e’lon qilish va o‘zgaruvchiga qiymat berish

Ushbu turdagi ma’lumotlar qiymatlari deklaratsiya bayonotidagi o‘zgaruvchilarga bevosita tayinlanadi.

Misol.



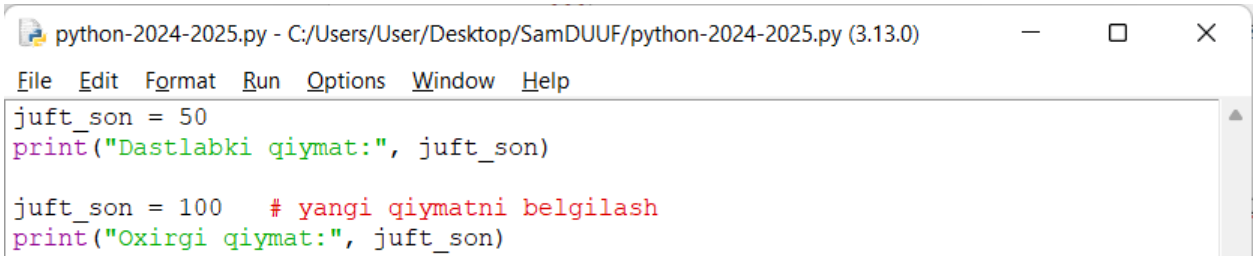
Natija.



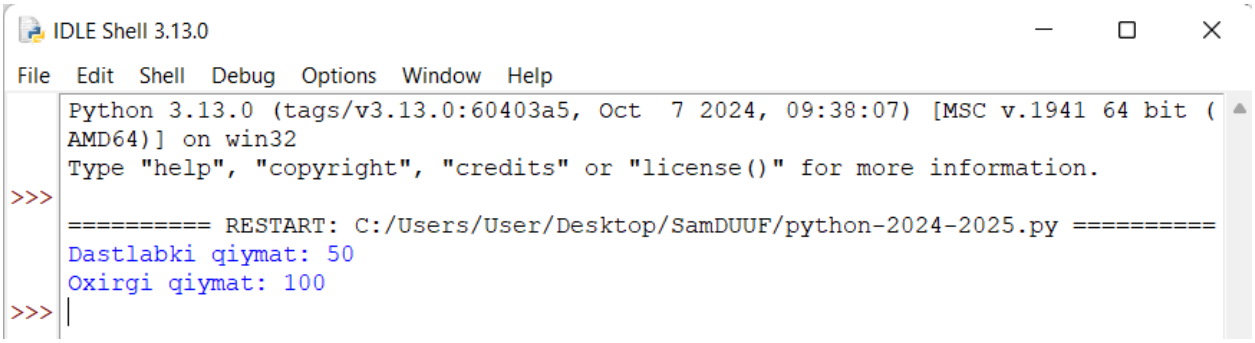
Bu yerda biz ‘son’, ‘sonlar qatori’ va ‘str’ deb nomlangan 3 ta o‘zgaruvchi yaratdik.

O‘zgaruvchining qiymatini o‘zgartirish

O‘zgaruvchilarga tayinlangan ma’lumotlar qiymatlari istalgan vaqtda o‘zgartirilishi mumkin. Python tilida siz o‘zgaruvchini narsalarni saqlash uchun sumka deb o‘ylashingiz mumkin va bu narsalarni istalgan vaqtda almashtirish mumkin.Misol.



Natija.



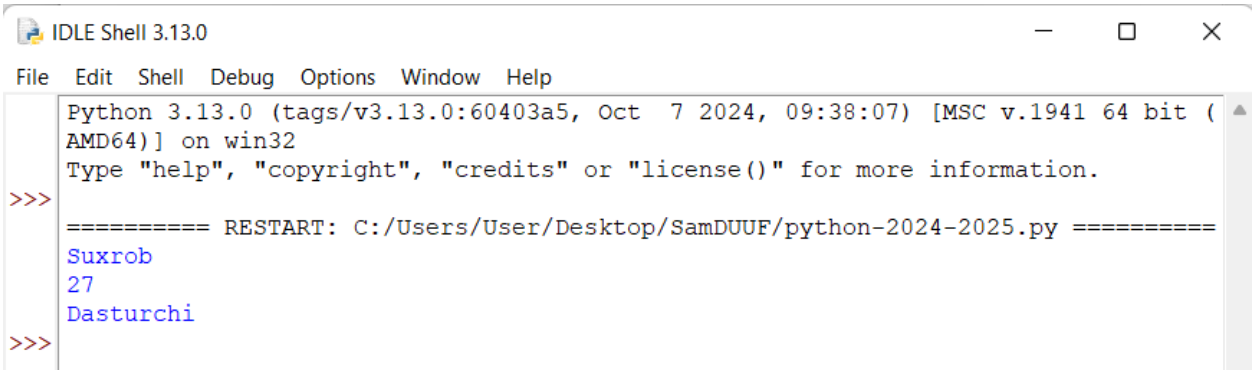
Yuqoridagi dasturda dastlab "juft_son" o‘zgaruvchisining qiymati 50 ga teng bo‘lgan. Keyinchalik u 100 qiymatiga qayta tayinlangan.

Bir nechta o‘zgaruvchilarga bir nechta qiymatlarni tayinlash

Python-da biz quyidagi usul yordamida bir xil deklaratsiya bayonotida bir nechta o‘zgaruvchilarga bir nechta qiymatlarni belgilashimiz mumkin

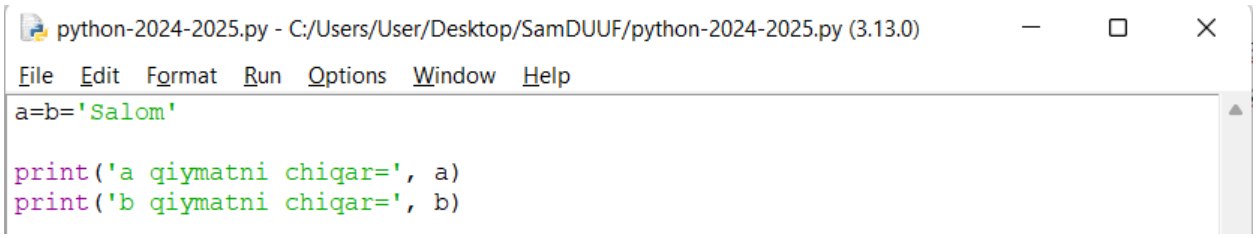
Misol

Natija

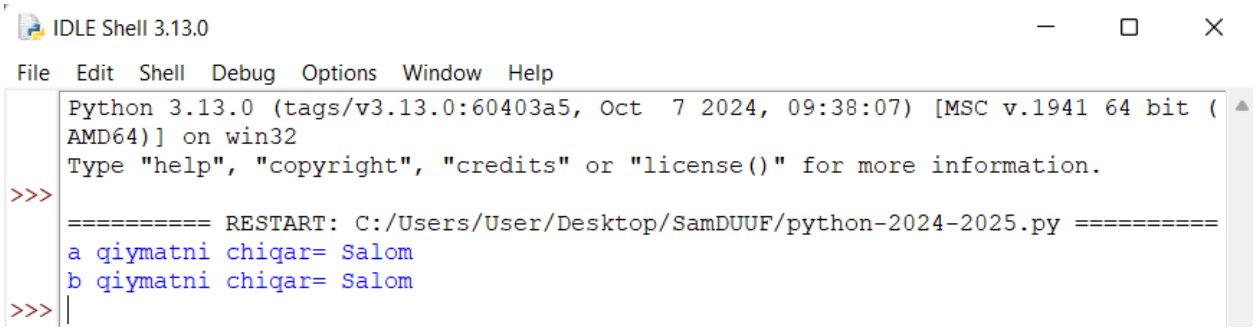


Bundan tashqari, bir nechta o‘zgaruvchilarga bitta qiymat belgilashimiz mumkin. Buni tushunish uchun quyidagi misolni ko‘rib chiqaylik.

Misol



Natija



Pythonda doimiylar: *Python Constant - bu butun dastur davomida qiymatini o‘zgartirib bo‘lmaydigan o‘zgaruvchi.*

Ba’zi qadriyatlar qat’iydir va haqiqat ekanligi umumjahon tomonidan tasdiqlangan. Ushbu qiymatlarni vaqt o‘tishi bilan o‘zgartirib bo‘lmaydi. Bunday turdagi qiymatlar doimiylar deb ataladi.

Eslatma - Boshqa dasturlash tillaridan farqli o‘laroq, Python hech qanday konstantalarni o‘z ichiga olmaydi. Buning o‘rniga, Python bizga katta harflar bilan yozilgan nomlash usulini taqdim etadi. Katta harf bilan yozilgan har qanday o‘zgaruvchi Pythonda doimiy deb hisoblanadi.

Konstantni e’lon qilishda amal qilish kerak bo‘lgan qoidalar

1. Python Konstantalari va o‘zgaruvchilar nomlari kichik (az) yoki bosh (AZ) belgilar, raqamlar (0-9) yoki pastki chiziq ( ) kombinatsiyasidan iborat bo‘lishi kerak.
2. Doimiy nomdan foydalanilganda, har doim KATTA HARF dan foydalaning, Masalan, CONSTANT = 50.
3. Doimiy nomlar raqamlar bilan boshlanmasligi kerak.
4. Pastki chiziqdan (_) tashqari, doimiyni e'lon qilishda qo‘shimcha maxsus belgilar (!, #, ^, @, $) ishlatilmaydi.

Konstantalarni ishlatish uchun albatta mavjud bo‘lgan kutubxonoani import(chaqirib) ishlatishimiz zarur. Misol uchun:

```
>>>from math import*
>>>print(pi)
3.141592653589793
>>>print(tau)
6.283185307179586
>>>print(e)
2.71828182845904
```

Nazorat savollari?

1. Pythonda doimiylar bormi?
2. Pythonda konstantalar nima?
3. Pythonda doimiyni qanday yaratasiz?
4. Pythonda o‘zgaruvchi va doimiy o‘rtasidagi asosiy farq nima?
5. O‘zgaruvchi yoki doimiyni nomlashda qanday qoidalar bor?

Mustaqil ishlash uchun topshiriqlar

1. Dasturlash tili tarkibidagi o‘zgaruvchilarni e’lon qiling.
2. Dasturlash tili tarkibidagi o‘zgarmaslarni e’lon qiling.
3. IDLE muhiti boshlang’ich oynasida quyidagi ifodalarning qiymatini hisoblang
   a) 41115+25684-25262=?
   b) 9+2*(-3)-6*(-2)=?
   c) 6.8*0.4*1.16/2.7*5.1*0.14=?
4. Python dasturlash tilida Salom Dunyo so‘zini chop eting.
5. Uch marta chiqarish operatoridan foydalangan holda, ism, familiya va sharifingizni ekranga chop eting.
6. a va b o‘zgaruvchilardagi qiymatlarni almashtiradigan dastur tuzing.
7. “ Salom” satr boshidagi probel belgilarni olib tashlash.

2.2-§. Python dasturlash tilida chiziqli jarayonlarni dasturlash.

Reja:

1. Python tilida chiziqli dasturlar;

2. type() va help() funksiyalari.

***Tayanch so‘zlar**: type, help, operator, chiziqli dastur, o‘zgaruvchi.*

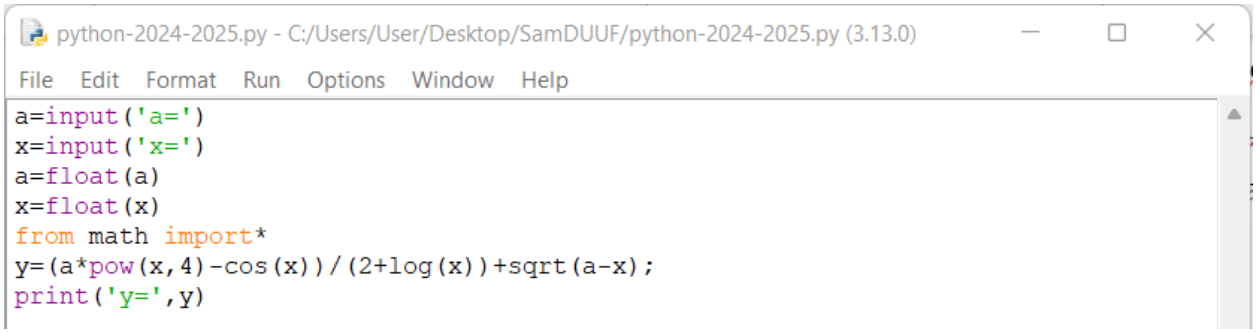
Programmalash tilida tuzilgan dasturlar odatda uchta jarayonga asoslanib tuziladi. Dasturlash tili operatorlari yechilayotgan masala algoritmini amalga oshirish uchun ishlatiladi. Operatorlar chiziqli va boshqaruv operatorlariga bo‘linadi. Aksariyat boshqa dasturlash tillarida operatorlar nuqtali vergul (‘;’) belgisi bilan tugallanadi python dasturlash tilida esa ; nuqtali vergulni quyish shart emas. Dastur tuzish vaqtida buyruqlar ketma-ketligi uzluksiz bajarilib boshqa shartlar talab etilmasa, dastur chiziqli hisoblanadi.

Tarif: Chiziqli algoritmlarga asoslanib python dasturlash tilida tuzilgan dasturlar chiziqli dasturlar deyiladi.

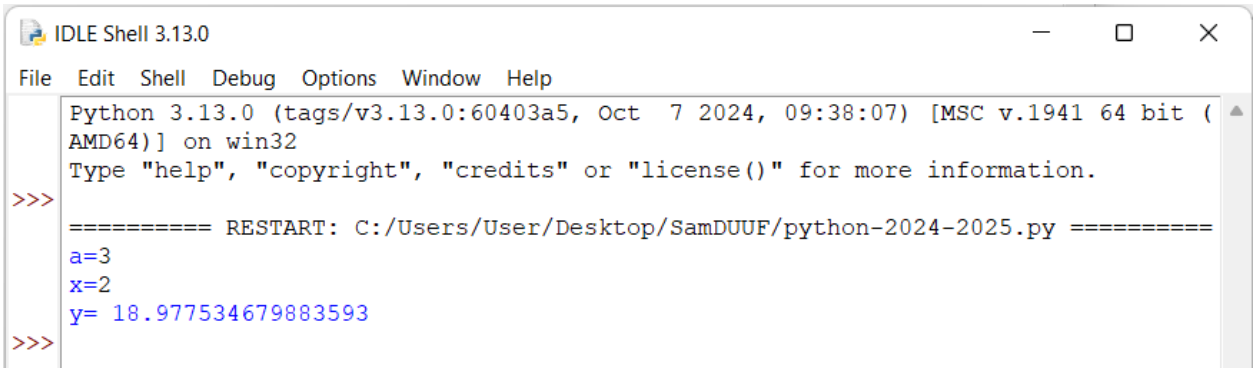
Chiziqli dasturlar tarkibiy qismi bo‘lgan operator va buyruqlarda hech qanday shart yoki takrorlanish bajarilmaydi. Chiziqli dasturlar tarkibidagi bo‘yruqlar, albatta, bir marta bajariladi.

Misol: Quyidagi funksiyani hisoblang ((a-x)>0).

$$y = \frac{ax^4 + \cos x}{2 + \ln x} + \sqrt{a - x}$$



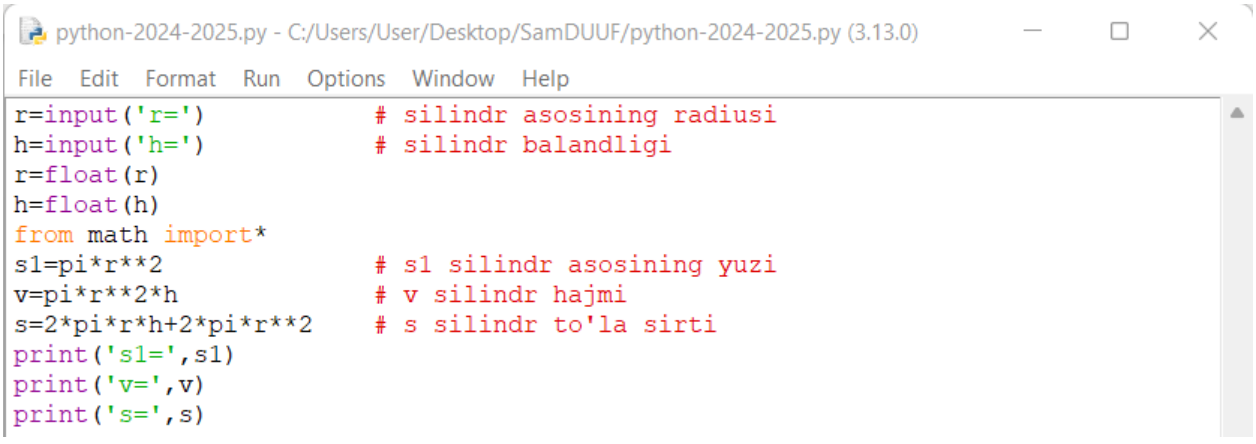
Natija.



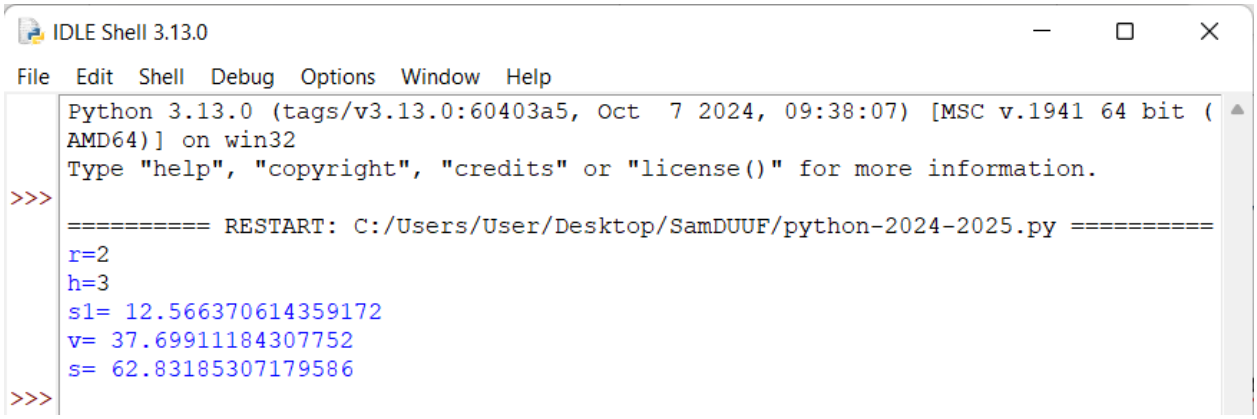
Yuqoridagi masalaga e'tibor bersak *a* va *x* o'zgaruvchilar qiymati berilganda y funksiyani natijasi hisoblandi, algoritm dastur tarkibidagi operatorlar ham bir marta bajarilyapti.

Misol: *Asosining radiusi r va balandligi h* bo'lgan *slindr asosining yuzi, hajmi va to'la sirtini toping.*

Bu masala yechimini aniqlash uchun silindr asosining yuzi, to'la sirti va hajm formulalarini aniqlash kerak. Berilgan r va h yordamida silindr asosining yuzi, to'la sirti va hajmini aniqlash mumkin.



Natija.



Python dasturlash tilida chiziqli dasturlar tuzilganda uning tarkibida matematik funksiyalar ishtirok etsa, albatta, matematik funksiyalar paketini chaqirish kerak. Ifodalarni ketma-ket ijro etish strukturasi Python tomonidan ta'minlanadi. Normal sharoitda python ifodalari dasturdagi buyruqlar yozilishiga ko‘ra bajariladi.

type() va help() funksiyalari

Dastur bajarilish vaqtida pythonda o‘zgaruvchilar turi e’lon qilinmasligi sababli ba’zi o‘zgaruvchilar natijasi qanday turda ekanligini aniqlash kerak bo‘ladi. Python dasturlash tilida o‘zgaruvchilar turini aniqlash uchun type() funksiyasi aniqlangan, type() funksiyasining umumiy ko‘rinish quyidagicha.

type(<o‘zgaruvchi>)

type(<o‘zgaruvchi >) – bu funksiya o‘zgaruvchining qanday turda ekanligini aniqlab beradi. Type funksiyasini ishlash jarayoni tushunarli bo‘lishi uchun, ularni interaktiv rejimda sinab ko‘ramiz. Chunki interaktiv rejim bir vaqtning uzida natija qaytaradi.

Misol. Type funksiyasini bajarilishi

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> a=12
>>> b=21.536
>>> c='matematika'
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(c)
<class 'str'>
>>>
```

Dastur tuzuvchiga dasturlash tili tarkibidagi yordam bo‘limi juda katta yordam beradi. Python dasturlash tilida funksiyalar qanday vazifani amalga oshirishni aniqlash uchun help() funksiyasi aniqlangan, help() funksiyasining umumiy ko‘rinish quyidagicha.

help(<funksiya>)

help(<funksiya>) – bu funksiya funksiyaning qanday qanday vazifa bajarishini aniqlab beradi.

Help funksiyasini ishlash jarayoni tushunarli bo‘lishi uchun, ularni interaktiv rejimda sinab ko‘ramiz.

Misol. Help funksiyasini bajarilishi

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> help(max)
Help on built-in function max in module builtins:

max(...)
    max(iterable, *[, default=obj, key=func]) -> value
    max(arg1, arg2, *args, *[, key=func]) -> value

    With a single iterable argument, return its biggest item. The
    default keyword-only argument specifies an object to return if
    the provided iterable is empty.
    With two or more arguments, return the largest argument.

>>>
```

Nazorat savollari

1.Chiziqli dasturlar deb nimaga aytiladi?

2.type() va help() funksiayalarning vazifalari?

Mustaqil ishlash uchun topshiriqlar

1. Nolga teng bo‘lmagan ikkita son berilgan. Ularning yig‘indisi, ko‘paytmasi va har birining moduli aniqlansin.

2. Uzunlik L santimetrda berilgan. Undagi to‘liq metrlar sonini aniqlovchi dastur tuzilsin (1 metr = 100 sm).

3. Og‘irlik M kilogramda berilgan. Undagi to‘liq tonnalar sonini aniqlovchi dastur tuzilsin (1 tonna = 1000 kg).

4. Faylning hajmi baytlarda berilgan. Bo‘lib butunni olish operatsiyasidan foydalanib fayl hajmining to‘liq kilobaytlarda ifodalovchi dastur tuzilsin (1 Kilobayt = 1024 bayt).

5. A va B (A > B) musbat sonlar berilgan. A kesmada, B kesmani necha marta joylashtirish mumkinligini aniqlovchi dastur tuzilsin.

6. A va B (A > B) musbat sonlar berilgan. A kesmada, B kesmani necha marta joylashtirish mumkin. A kesmada B kesmaning joylashmagan qismini aniqlovchi dastur tuzilsin.

7. Ikki xonali son berilgan. Oldin uning o‘nliklar xonasidagi raqamni, So‘ng birlar xonasidagi raqamni chiqaruvchi dastur tuzilsin.

8. Ikki xonali son berilgan. Uning raqamlar yig‘indisi va ko‘paytmasini aniqlovchi dastur tuzilsin.

9. Ikki xonali son berilgan. Uning raqamlari o‘rnini almashtirishdan hosil bo‘lgan sonni aniqlovchi dastur tuzilsin.

10. Uch xonali son berilgan. Uning yuzlar xonasidagi raqamini aniqlovchi dastur tuzilsin.

11. 999 dan katta bo‘lgan son berilgan. Bir marta bo‘lib butunni va bo‘lib qoldiqni olish operatsiyasidan foydalanib berilgan sonni mingliklar xonasidagi sonni aniqlovchi programma tuzilsin.

12. Kun boshidan boshlab N sekund vaqt o‘tdi. Kun boshidan boshlab qancha minut to‘la o‘tganligini aniqlovchi programma tuzilsin.

13. Hafta kunlari quyidagi tartibda berilgan: 1 – dushanba, 2 – seshanba, ... , 6 – shanba, 7 – yakshanba (N 1-7gacha bo‘lgan hafta kunlari soni). 365 oraliqda yotuvchi K soni berilgan. Agar 1-yanvar N chi kunga to‘g‘ri kelsa, kiritilgan K – kun haftaning qaysi kuniga to‘g‘ri kelishini aniqlovchi programma tuzilsin.

14. A, B, C butun sonlar berilgan. Tomonlari AxB bo‘lgan to‘g‘ri to‘rtburchakka tomoni C bo‘lgan kvadrat eng ko‘p joylashtirilsin. To‘g‘rito‘rt burchakka eng ko‘p joylashgan kvadratlar soni va joylashmay qolgan qismi yuzasini aniqlovchi programma tuzilsin.

15. Qaysidir yil berilgan. Berilgan yilning qaysi yuz yillikka kirishini aniqlovchi programma tuzilsin. (Masalan: 20 – yuz yillikning boshi 1901 yil).

2.3-§. Python dasturlash tilida tarmoqlanuvchi jarayonlarni dasturlash.

Reja:

1. *Qisqa shartli operator va uning umumiy ko‘rinishi;*
2. *To‘liq shartli operator va uning umumiy ko‘rinishi;*
3. *elif operatori va uning umumiy ko‘rinishi.*

Tayanch so‘zlar. *If, else, elif, qisqa shartli, to‘liq shartli.*

Python dasturlash tilida chiziqli jarayonlar buyruqlar ketma-ketligi asosida bajariladi, tarmoqlanuvchi jarayonlarni dasturlashda esa, buyruqlar ma’lum bir shartlar asosida tarmoqlanish bo‘yicha bajariladi. Python dasturlash tilida tarmoqlanuvchi(shartli) jarayonlarni bir necha turlarga bo‘lingan holda amalga oshirish mumkin.

Qisqa shartli operator va uning umumiy ko‘rinishi

Tarmoqlanuvchi jarayonlarni amalga oshiruvchi operatorlarni qisqacha qilib shartli operatorlar deb yuritamiz. Shartli operatorlar ham qisqa, to‘liq va umumiy shartli operator ko‘rinishlarida, tarmoqlanishni amalga oshiradi.

if operatori

Tarmoqlanuvchi jarayonlarni python dasturlash tilida amalga oshirish uchun, albatta, tarmoqlanuvchi algoritmlar asosida bajariladi. Algoritm bajarilish vaqtida ma’lum bir shartlar asosida algoritmning u yoki bu qismi bajarilishini ta’minlash maqsadida shartli operatorlardan foydalaniladi.

Tarif: Algoritm tarkibidagi shart asosida algoritmning tarmoqlarga bo‘linishiga xizmat qiluvchi operatorlar shartli operatorlar deyiladi.

Hayotdagi asosiy ko‘p masalalarni dasturlash vaqtida, albatta, tarmoqlanuvchi algoritmlar asosida bajariladi. Tarmoqlanuvchi algoritmlar ham ikki xil holatni o‘z ichiga oladi, ya’ni shart rost bo‘lganda ma’lum bir vazifani yolg‘on bo‘lganda hech qanday vazifani bajarmaydi va shart rost bo‘lganda ma’lum bir vazifani yolg‘on bo‘lganda boshqa vazifani bajaradi. Demak, yuqoridagi ikki holatni e’tiborga olib, shartli operatorlar ham qisqa va to‘liq ko‘rinishga ega.

Qisqa shartli operatorning umumiy ko‘rinishi quyidagicha

```
if <mantiqiy ifoda> :
    operatorlar
```

if operatori tarkibida shartlar ikki va undan ortiq bo‘lsa oldingi bobdagi mantiqiy ifodalar asosida birlashtiriladi. if operatori tarkibidagi shart faqat chin bo‘lgandagina : dan keyingi opertorlar bajariladi.

Misol*: Berilgan sonning juft yoki toq ekanligini aniqlang.*

```
 a=10
if a%2==0:
   print('juft')
if a%2!=0:
   print('toq')
========== RESTART: C:\Users\User\Desktop\2.py
==========
juft
>>>
```

Berilgan masala yechimida faqat bitta shart tekshirildi, bu yerda blok yoki begin end vazifalari hech qanday buyruqsiz amalga oshiriladi. Agar shartdan keyin bir nechta operator bajarilish uchun **:** belgisidan keyin ENTER tugmasi bosilib yozilaveradi yani bir soha bo‘yicha.

***Misol**: Berilgan a va b sonda a>b bo‘lsa ikkita sonning yig‘idisi va ayirmasini, a<=b bo‘lsa ikkita sonning ko‘paytmasi va bo‘linmasini toping.*

```
a=10
b=5
if a<=b:
   y=a+b
   z=a-b
if a>b:
   y=a*b
   z=a/b
print('y=',y,' z=',z)
========== RESTART: C:\Users\User\Desktop\2.py
==========
y= 50  z= 2.0
>>>
```

To‘liq shartli operator va uning umumiy ko‘rinishi

Python dasturlash tilida tarmoqlanuvchi jarayonlarni to‘liq shartli ko‘rinishida ifodalash uchun **if else** operatoridan foydalaniladi. Algoritm

tarkibidagi shartlar chin qiymat qabul qilganda ma’lum bir operatorlar, yolg‘on qiymat qabul qilganda boshqa operatorlar bajarilishi ham mumkin.

Tarmoqlanuvchi jarayonlarni python dasturlash tilida to‘liq shartli operatorning umumiy ko‘rinishi quyidagicha

if <mantiqiy ifoda> :

operatorlar1

else:

operatorlar2

if else operatori tarkibidagi shartlarning chin qiymat qabul qilganda operatorlar1 bajariladi, aks holda operatorlar2 bajariladi.

Misol: Berilgan son juft bo‘lsa 2 ga ko‘paytirib juft so‘zini aks holda toq bo‘lsa 3 ga ko‘paytirib toq so‘zini ekranga chop eting.

```
a=15
if a%2==0:
 print('juft')
 print(2*a)
else:
 print('toq')
 print(3*a)
============== RESTART: C:\Users\User\Desktop\2.py
==============
toq
45
>>>
```

Misol. *Pythonda quyidagi masalani dasturiy ta’minotini yarating.*

$$f(x)=\begin{cases}0.5\cdot\cos(x), & agar\ \ x>0;\\ x+8, & agar\ \ x\le 0;\end{cases}$$

```
x=float(input('x='))
from math import*
if x>0:
 y=0.5*cos(x)

else:
 y=x+8
print('y=',y)
============ RESTART: C:\Users\User\Desktop\2.py
================
x=-2.8
y= 5.2
>>>
```

elif operatori

Tarmoqlanuvchi jarayonlarni algoritmlash vaqtida, albatta, shartga e’tibor berish talab etiladi, chunki algoritm shart bo‘yicha tarmoqlanib qismlarga ajralib ketadi. Masalan, dastur bajarilishining birorta qadamida qandaydir shartni tekshirish natijasiga ko‘ra boshqaruvni dasturning u yoki bu bo‘lagiga uzatish mumkin.

elif shart operatori Python dasturlash tilida aniqlangan bo‘lib, u aks holda ichida shart kelgan holatlarda ishlatiladi. **Elif shart operatori**ning umumiy ko‘rinishi quyidagicha.

if <mantiqiy ifoda1> :

operatorlar1

elif <mantiqiy ifoda2> :

operatorlar2

else:

operatorlar3

Bu operator aks holda qismida alohida if operatorini yozmaslik uchun ishlatiladi, agar <mantiqiy ifoda1> false qiymatidan farqli ya’ni true bo‘lsa, <operatorlar1> bajariladi, aks holda <mantiqiy ifoda2> tekshiriladi, agar chin bo‘lsa <operatorlar2> bajariladi, aks holda <operatorlar3> bajariladi.

Misol*. Berilgan musbat sonni kabisa yili ekanligini aniqlang. 4 ga karrali yillar va 100 ga karralilar ichida faqat 400 ga karralilari Kabisa yili hisoblanadi.*

Masalan: 16,24,2016,2020, 2400 lar kabisa va 5,100,200, 2017 lar kabisa yili emas.

```
x=float(input('x='))
from math import*
if x%100==0:
 if x%400==0:
 print('kabisa yili')
 else:
```

```
 print('kabisa yili emas')
elif(x%4==0):
 print('kabisa yili')
else:
 print('kabisa yili emas')
========= RESTART: C:\Users\User\Desktop\2.py
====================
x=2020
kabisa yili
>>>
```

Tarmoqlanuvchi jarayonlarini algoritmlarida uchraydigan shartlar holatiga qarab, yuqorida keltirib o‘tilgan uch xil ko‘rinishdagi shartli operatorlarning qulay birortasidan foydalaniladi.

Nazariy savollar.

1. Tarmoqlanuvchi algoritmlar?
2. Tarmoqlanuvchi jarayonlarni dasturlashning necha xil usuli mavjud?
3. To‘liq shartli operatorning ta’rifi, uning umumiy ko‘rinishi va vazifasi?
4. Qisqa shartli operatorning ta’rifi, uning umumiy ko‘rinishi va vazifasi?
5. Umumiy shartli operatorning ta’rifi, uning umumiy ko‘rinishi va vazifasi?

Mustaqil ishlash uchun topshiriqlar

1. Uchta butun son berilgan. Shu sonlarning ikkitasi o‘zaro teng, qolgan bittasini tartib raqami aniqlansin.
2. To‘rtta butun son berilgan. Shu sonlarning uchtasi o‘zaro teng, qolgan bittasini tartib raqami aniqlansin.
3. Sonlar o‘qida uchta A, B, C nuqtalar berilgan. A nuqtaga eng yaqin nuqta va ular orasidagi masofa topilsin.
4. Ikkita A va B butun tipli o‘zgaruvchilar berilgan. Agar ularning qiymatlari o‘zaro teng bo‘lmasa, har bir o‘zgaruvchiga bu qiymatlar yig‘indisi ta’minlansin, agar o‘zaro teng bo‘lsa, o‘zgaruvchilarga 0 ta’minlansin. A va B ning yangi qiymatlari ekranga chiqarilsin.
5. Uchta son berilgan. Ularning o‘rtasidagi (ya’ni kattasi va kichigi orasida joylashgan) sonni aniqlovchi dastur tuzilsin.
6. Uchta son berilgan. Shu sonlarning yig‘indisi eng katta bo‘ladigan ikkitasini ekranga chiqaradigan dastur tuzilsin.
7. Kordinatalar tekisligida butun son berilgan. Agar nuqta koordinata boshida yotsa, 0 chiqarsin. Agar nuqta OX yoki OY o‘qlarida joylashsa mos holda 1 va 2 chiqarilsin. Agar nuqta koordinata o‘qida joylashmasa 3 chiqarilsin.
8. OX va OY koordinata o‘qlarida yotmaydigan nuqta berilgan. Nuqta joylashgan koordinata choragi aniqlansin.
9. Koordinata o‘qlariga parallel ravishda to‘g‘ri to‘rtburchakning uchta uchi berilgan, to‘rtinchi uchi koordinatasini aniqlansin.
10. Yil berilgan (musbat butun son). Berilgan yilda nechta kun borligini aniqlovchi dastur tuzilsin.

11. Butun son berilgan. Berilgan sonni “musbat toq son”, “manfiy juft son”, “son nolga teng” va h.k ekranga yozadigan dastur tuzilsin.

12. 1-999 oralig‘idagi sonlar berilgan. Berilgan sonni “ikki xonali juft son”, “uch xonali toq son” va h.k ekranga yozadigan dastur tuzilsin.

2.4-§. Takrorlanuvchi jarayonlarni dasturlash, for(sikl) operatori va uning umumiy ko‘rinishi.

Reja:

1. *Takrorlanuvchi jarayonlarni dasturlash;*
2. *for(sikl) operatori va uning umumiy ko‘rinishi;*

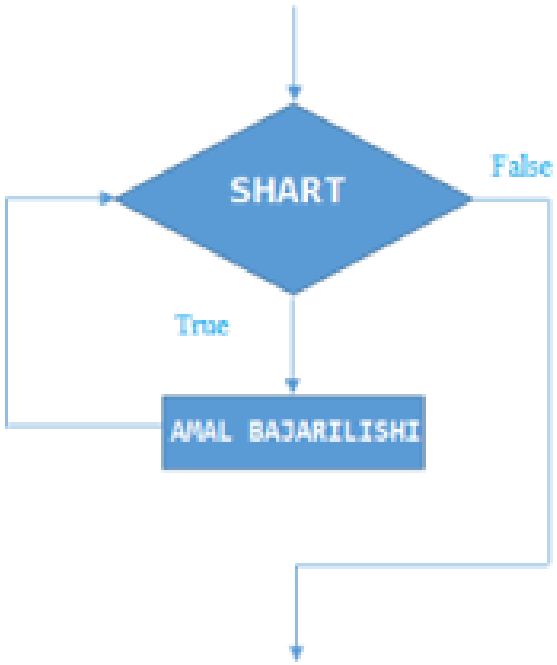
Tayanch so‘zlar. *For, sikl, takrorlanish, parametrli takrorlanish.*

Dastur tuzish jarayonida ba’zi bir masalalarni algoritmlari tarkibidagi buyruqlar ikki va undan ortiq marta bajarilishiga to‘g‘ri keladi. Agar algoritm tarkibidagi bir necha marta takrorlanishi kerak bo‘lgan buyruqlarni takrorlanuvchi jarayonlar asosida dasturlash tillarida tasvirlanmasa, bu buyruqlarni barchasini bajarish murakkablashadi. Elektron hisoblash mashinalarini insoniyatdan farqi shundaki, insoniyatda bir nechta buyruqlarni bajarish davomida toliqish holatlari bo‘lishi mumkin, elektron hisoblash mashinalariga takrorlanishni qanchaligini ma’lum bir buyruqlar asosida berilsa, ular barchasini charchamasdan bajaradi. Ba’zi bir takrorlanuvchi jarayonlarni, takrorlanish formulasini chiqarib oddiy hisoblash mumkin, lekin ixtiyoriy ketma ketliklar yig‘indisini hisoblash oddiy usullar bilan hal etilmaydi, bunday holatlarda takrorlanuvchi jarayonlardan foydalaniladi.

Tarif: Algoritmning qandaydir qismidagi buyruqlar ikki va undan ortiq bajarilishiga takrorlanuvchi jarayonlar deyiladi.

Yuqoridagi ta’rifga etibor qaratsak, demak algoritmning qandaydir qismi ikki va undan ortiq bajarilishi mumkin bo‘lgan holatlar ham mavjud. Bunda dasturchiga shunday vazifa qo‘yiladiki takrorlanish holatini bir yaxlit buyruq asosida kompyuterga qulay usulda berish kerak.

Takrorlanuvchi jarayonlarni quyidagi blok sxema ko‘rinishda ixtiyoriy dasturlash tilida tasvirlash mumkin.



Yuqoridagi blok sxema shaklida shart toki chin bo‘lgunga qadar takrorlanish bajarilaveradi, aks holda takrorlanish to‘xtatiladi.

Python dasturlash tillarida takrorlanuvchi jarayonlarni quyidagi usullar yordamida tasvirlash mumkin.

- Parametr bo‘yicha takrorlash(for);
- Shart bo‘yicha takrorlash(while, do while).

for(sikl) operatori

Takrorlanuvchi jarayonlarni takrorlanish soni aniq bo‘lgan holatlardagina parameter bo‘yicha takrorlash usulidan foydaliniladi. Takrorlanuvchi jarayonlarni parametr bo‘yicha Python dasturlash tilida tasvirlash uchun, albatta, takrorlanish soniga e’tibor berish kerak. Parametr bo‘yicha takrorlanuvchi jarayonlarga, masalan, birdan n gacha sonlarning kvadratlarini yig‘indisini topish kabi misollar kiradi. Bundatakrorlanish soni aniq, ya’ni birdan dan n gacha deb berilyapti.

Parametr bo‘yicha takrorlanuvchi jarayonlar takrorlanish oshishi yoki kamayishiga qarab ikki turga bo‘linadi:

- noldan boshlab qadam 1 ga teng bo‘lgan takrorlanish(1-tur);
- a dan boshlab qadam 1 ga teng bo‘lgan takrorlanish(2-tur);
- a dan boshlab b gacha qadam x ga teng bo‘lgan takrorlanish(3-tur).

Takrorlanuvchi jarayonlarni dasturlash vaqtida takrorlanish qadami birga oshib borilsa, birinchi turdan foydaliniladi. 1-tur takrorlanish qadami noldan boshlanib birga oshib boruvchi parameter bo‘yicha sikl operatorining umumiy ko‘rinishi quyidagicha.

for <*o‘zgaruvchi1*> in range(<*o‘zgaruvchi2*>) :

operatorlar

1-tur for operatori takrorlanish sonining boshlang‘ich va oxirgi qiymatlari aniq bo‘lgandagina ishlatiladi. For operatorining ishlash prinsipi takrorlanish *<o‘zgaruvchi1> = 0* dan boshlanib toki *<o‘zgaruvchi1> = <o‘zgaruvchi2 - 1>* ga teng bo‘lguncha davom etadi, bunda har bir qadamda <o‘zgaruvchi1> ni qiymati birga oshib boradi. Bu yerda range() funksiyasi takrorlanish qadami va oxirini ta’minlab berishga xizmat qiladi.

Misol. 1 dan n gacha sonlarning kublari yig‘indisini hisoblash dasturini tuzing.

```
n=input('n=')
n=int(n)
s=0
for i in range(n):
 s+=(i+1)**3
print('s=',s)
========== RESTART: C:\Users\User\Desktop\1.py
================
n=100
s= 25502500
>>>
```

Misol: Quyidagi yig‘indini hisoblash dasturini tuzing.

$$s = \frac{cos2}{1} + \frac{cos^n 2}{2} + \frac{cos^3 2}{6} + \cdots + \frac{cos^n 2}{n!}$$

Bu masalani hisoblash jarayonida sikl operatori tarkibida yig’indini suratini hisoblash uchun alohida funksiya, maxrajini hisoblash uchun alohida fuksiya va yig‘indi uchun alohida funksiya yaratib olsak maqsadga muvofiq bo‘ladi.

```
n=input('n=')
n=int(n)
p=1
s=0
t=1
from math import*
for i in range(n):
```

```
 p*=(i+1)
 t*=cos(2)
 s+=t/p
print('s=',s)
60
======== RESTART: C:\Users\User\Desktop\1.py
=================
n=15
s= -0.340416587577421 16
>>>
```

Takrorlanuvchi jarayonlarni dasturlash vaqtida takrorlanish qadami a dan boshlanib qadam 1 bilan b gacha bajarilish ham mumkin, bunda, ikkinchi turdan foydaliniladi. 2-tur takrorlanish qadami a dan boshlanib qadam 1 ga oshib boruvchi parametr bo‘yicha sikl operatorining umumiy ko‘rinishi quyidagicha.

for <*o‘zgaruvchi1*> **in range(**<*o‘zgaruvchi2*>**,** <*o‘zgaruvchi3*>**) : operatorlar**

Bunda for operatorining boshlang‘ich qiymati **<o‘zgaruvchi2>** dan boshlanadi. For operatorining ishlash prinsipi **<o‘zgaruvchi1> = <o‘zgaruvchi2>** dan takrorlanish boshlanib toki **<o‘zgaruvchi1>=<o‘zgaruvchi3 - 1>** gacha davom etadi, **<o‘zgaruvchi1>**ni qiymati birga oshib borishi ta’minlaydi.

Misol*. a dan b gacha sonlar orasida toqlarining kvadratlar, juftlarini kublar yig‘indisini toping.*

```
a=input('a=')
a=int(a)
```

```
b=input('b=')
b=int(b)
p=0 # toqlar yig‘indisi
s=0 # juftlar yig‘indisi
from math import*
for i in range(a,b+1):
 if i%2==0:
    s+=i**3
 else:
    p+=i**2
print('Juftlar kublar yigindisi=',s)
print('Toqlar kvadratlar yigindisi=',p)
==================== RESTART: D:\python dasturlari\juft,toq.py ====================
a=1
b=3
Juftlar kublar yigindisi= 8
Toqlar kvadratlar yigindisi= 10
```

Takrorlanishni uchinchi turi a dan boshlab b gacha qadam x ga teng bo‘lgan takrorlanish jarayonlarini dasturlashni qarab o‘tamiz. Bunda takrorlanish ixtiyoriy intervalda ixtiyoriy qadam bilan amalga oshirilish mumkin. C++, Pascal tillarida -1 qadam alohida yozilar edi, python tilida esa manfiy va musbat qadamlar uchinchi tur bo‘yicha hal etiladi.

Takrorlanish a dan boshlab b gacha qadam x ga teng bo‘lgan parametr bo‘yicha sikl operatorining umumiy ko‘rinishi quyidagicha.

for *<o‘zgaruvchi1>* **in range(***<o‘zgaruvchi2>*,

<o‘zgaruvchi3>,<o‘zgaruvchi4>) :

operatorlar

Bunda for operatorining boshlang‘ich qiymati **<o‘zgaruvchi2>** dan boshlanadi. Yuqoridagi operatorning ishlash prinsipi **<o‘zgaruvchi1> = <o‘zgaruvchi2>** dan takrorlanish boshlanib toki **<o‘zgaruvchi1>=<o‘zgaruvchi3 - 1>** gacha davom etadi, **<o‘zgaruvchi1>**ni qiymati har takrorlanishda **<o‘zgaruvchi4>**ga oshib borishi ta’minlanadi.

Misol*. a dan b gacha juft sonlarni to‘rtinchi darajalar yig‘indisini hisoblash dasturini tuzing.*

```
a=input('a=')
a=int(a)
b=input('b=')
b=int(b)
if a%2!=0:
      a+=1
s=0
from math import*
for i in range(a,b+1,2):
      s+=i**4


print('s=',s)
```

```
======== RESTART: C:\Users\User\Desktop\1.py
=================
a=1
b=5
s= 272
>>>
```

Yuqoridagi misolda a sonini toq ekanligi aniqlanadi, for tarkibida range() funksiyasi b-1 gacha bajarilishi sababli, biz b+1ni yozamiz. For operatori takrorlanishni a dan b gacha 2 qadam bilan amalga oshiradi.

Uchinchi tur bo‘yicha for sikl operatori yordamida katta sondan kichik songacha manfiy qadam bilan ham takrorlanishni amalga oshirish mumkin.

Misol. *a dan b (a>b) gacha sonlarni ikkiga ko‘paytirib ekranga chiqarish dasturini tuzing.*

```
a=input('a=')
a=int(a)
b=input('b=')
b=int(b)
for i in range(a,b-1,-1):
 print(2*i)
======== RESTART: C:\Users\User\Desktop\1.py
=================
a=1
b=-10
2
```

```
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
>>>
```

Nazariy savollar

1. Takrorlanuvchi jarayonlar deb nimaga aytiladi?
2. Takrorlanuvchi jarayonlarni blok sxema ko‘rinishida tasvirlanishi?
3. Takrorlanuvchi jarayonlarni dasturlash tilida tasvirlanish usullari?
4. For sikl operatori va uning vazifasi?
5. For sikl operatorining turlari?
6. For sikl operatorining umumiy ko‘rinishi?
7. 1 dan n gacha siklni tashkil qilishni tushuntirib bering?
8. a dan b gacha siklni tashkil qilishni tushuntirib bering?
9. a dan b gacha x qadamli siklni tashkil qilishni tushuntirib bering?

Mustaqil ishlash uchun savollar

1. A va B butun sonlari berilgan (A<B). A dan B gacha bo‘lgan barcha butun sonlar (A va B ham kiradi) kvadratlarining yig‘indisini chiqaruvchi dastur tuzilsin.

2. N (N>0) butun son berilgan. Quyidagi yig‘indini hisoblovchi dastur tuzilsin:

$S=1+1/2+1/3+ . . . + 1/N.$

3. N (N>0) butun son berilgan. Quyidagi yig‘indini hisoblovchi dastur tuzilsin:

$S=N^2+ (N+1)^2+(N+2)^2+ . . . + (2N)^2.$

4. N (N>0) butun son berilgan. Quyidagi ko‘paytmani hisoblovchi dastur

tuzilsin: P=1.1*1.2*1.3* ... (N ta ko‘paytuvchi).

5. N (N>0) butun son berilgan. Quyidagi ifodani hisoblovchi dastur tuzilsin: S=1.1-1.2+1.3- ...(N ta qo‘shiluvchi ishoralar almashib keladi. Shartli operatordan

foydalanmang).

6. N (N>0) butun son berilgan. Shu sonning kvadratini quyidagi formula asosida hisoblovchi dastur tuzilsin:$N^2= 1+3+5+ ... + (2N-1)$. har bir qo‘shiluvchidan keyin natijani ekranga chiqarib boring. Natijada ekranda 1 dan N gacha bo‘lgan sonlarning kvadratlari chiqariladi.

7. A xaqiqiy va N (N>0) butun sonlari berilgan. A ning N-darajasini aniqlovchi dastur tuzilsin: $A^n= A*A* … *A$

8. A xaqiqiy va N (N>0) butun sonlari berilgan. Bitta ssikldan foydalanib A ning 1 dan N gacha bo‘lgan barcha darajasini chiqaruvchi dastur tuzilsin

III BOB. PYTHON DASTURLASH TILIDA MURAKKAB TURLAR

3.1-§. Ro‘yxatlar.

Reja:

1. *Ro‘yxatlar haqida umumiy ma’lumotlar;*
2. *Ro‘yxatlarga doir dasturlar*;

Tayanch so‘zlar. *Ro‘yxat, len, max, min, sum, sorted, del, append.*

Pythonda ma’lumot to‘plamlarining turlari 4 xil bo‘lib ulardan odatda bir yoki undan ortiq qiymatlarni saqlashda foydalanamiz. Bizga kerak bo‘lganda o‘sha to‘plamlarga murojaat qilib tegishli qiymatlarni olamiz. Har bir ma’lumot to‘plamining o‘z xususiyatlari bor va shunga ko‘ra ularni kerakli joyda tanlab ishlatamiz. Darsimiz davomida barcha turlarning xususiyatlarini ko‘rib chiqamiz:

- List – tartiblangan va o‘zgaruvchan ro‘yxat. Elementlarini dublikatlash mumkin.
- Tuple – tartiblangan va o‘zgarmas ro‘yxat. Elementlarini dublikatlash mumkin.
- Set – Tartiblanmagan va indekslanmagan to‘plam. Elementlari dublikatlanmaydi.
- Dictionary – tartiblanmagan,o‘zgaruvchan va indekslangan to‘plam. Elementlari dublikatlanmaydi.

Yuqoridagi xususiyatlardan kelib chiqqan holda tegishli joylarda qo‘llaniladi. Ular bilan birma-bir tanishib chiqamiz.

Ta’rif: Har xil turga mansub bo‘lgan yagona no bilan saqlangan tartiblangan ma’lumotlar majmuasi ro‘yxat(lest) deb ataladi.

Ro‘yxatlar python dasturlash tilida [] to‘rtburchak qavs ichda elon qilinadi. Ro‘yxatlarning har bir elemantlariga indeks raqami orqali murojaat qilinadi.

Ro‘yxatlarni boshlang‘ich qiymatlarini bergan holatda ishga tushurish quyidagicha amalga oshiriladi.

<ro‘yxat o‘zgaruvchisi>=[qiymat1, qiymat2, qiymat3, …]

Ro‘yxatni python dasturlash tilida faol qilish uchun, albatta uning elementlari soni berilishi shart emas. Ro‘yxatning elementlari soni uning tarkibidagi qiymatlariga qarab aniqlanadi.

Python dasturlash tilida ro‘yxatlarni faollashtirish va ulardan foydalanish.

```
>>> q=[5,3,1,'s',7,9,10,11]
>>> q
[5, 3, 1, 's', 7, 9, 10, 11]
>>> q[1]
3
>>> q[4]
7
>>> q[3]
's'
>>> q[2]+q[7]
12
>>> S
```

Yuqoridagi dastur kodiga etibor qaratadigan bo‘lsak quyidagilarni aytishimiz mumkin: Ro‘yxat elementlari ixtiyoriy turda bo‘ladi;

- Ro‘yxat elementlari tartiblanmagan holda bo‘ladi;

- Ro‘yxat elementlari vergul bilan ajratiladi;
- Ro‘yxat elementlari ichida string turdagi ma’lumotlar kelsa bittalik tirnoq ichida yoziladi;
- Ro‘yxat elementlari 0-tartibdan boshlanadi;
- Ro‘yxat elementlariga indeks orqali murojaat qilinadi.

Manfiy indeks sanoq oxiridan boshlanishini bildiradi. Masalan, -1 eng oxirgi, -2 oxiridan oldingi element va hokozo.

Masalan: meva=['olma’, 'nok', "o‘rik"] ro‘yxatni oxirgi elementini ekranga chiqaramiz.

```
>>> meva=['olma', 'nok', "o'rik"]
>>> meva[-1]
"o'rik"
>>>
```

Ro‘yxat elementlari ustida amallar bajarish mumkin bo‘lib ular aniqlangan funksiyalar deb nomlanadi, bu funksiyalarni quydagi dastur orqali ko‘rib chiqamiz.

```
>>> a=[4,6,8,9,1,2,5,3,7,15,11,0]
>>> a
[4, 6, 8, 9, 1, 2, 5, 3, 7, 15, 11, 0]
>>> len(a)
12
>>> max(a)
15
>>> min(a)
0
>>> sum(a)
71
>>> sorted(a)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 15]
>>> del(a[3])
>>> a
[4, 6, 8, 1, 2, 5, 3, 7, 15, 11, 0]
>>>
```

List ro‘yxatidagi istalgan element qiymatini o‘zgartirish mumkin. Buning uchun uning indeksi orqali murojaat qilib, yangi qiymatni biriktiramiz. Hozir ro‘yxatdagi birinchi elementni o‘zgartiramiz:

```
>>> kitoblar=['matematika', 'informatika', 'ona tili', 'kimyo']
>>> kitoblar[2]='fizika'
>>> kitoblar
['matematika', 'informatika', 'fizika', 'kimyo']
>>>
```

Elementning mavjudligini tekshirish

Biror elementning ro‘yxatda mavjudligini tekshirish uchun in operatoridan foydalaniladi. Hozir ro‘yxatda 6 borligini tekshiramiz:

```
>>> sonlar=[1,5,9,4,3,2,7,6,10,15,0]
>>> 8 in sonlar
False
>>> 9 in sonlar
True
>>>
```

Ro‘yxatni yaratgandan so‘ng uning ustida turli amallarni bajarish kerak bo‘ladi, albatta, buning uchun esa Pythonni o‘ziga kiritilgan bir qancha funksiya va metodlar bor.

Metod	Vazifasi
List.append(x)	Ro`yxat oxiridan element qo`shish
List.extend(L)	Oxiriga hamma elementlarni qo`shib list ro`yxatini kengaytiradi.
List.insert(i,x)	i-elementga x qiymatini kiritadi
List.remove(x)	Ro`yxatdan x qiymatga ega elementni o`chiradi
List.pop([i])	Ro`yxatning i-elementini o`chiradi va qaytaradi. Agarda indeks ko`rsatilmagan bo`lsa oxirgi element o`chiriladi
List.index(x,[start],[end])	X qiymatga teng start dan end gacha birinchi elementni qaytaradi
List.count(x)	X qiymatga teng elementlar sonini qaytaradi
List.sort([key=funksiya])	Funksiya asosida ro`yxatni saralaydi
List.reverse()	Ro`yxatni ochadi
List.copy()	Ro`txatning nusxalaydi
List.clear()	Ro`yxatni tozalaydi

Nazariy savollar

1. Pythonda ma’lumot to‘plamlarining turlarini bilasizmi?
2. Ro‘yxat deganda nimani tushunasiz?
3. Ro‘yxat ustida qanday amallar bajarilishini bilasizmi?
4. Ro‘yxat elemantlariga qanday murojaat qilinadi?
5. Ro‘yxat elementlari qanday turdagi ma’lumotlardan tashkil topgan.

Mustaqil ishlash uchun topshiriqlar

1. Natural m hamda haqiqiy sonli A(1:m) ro‘yxat berilgan uning a:b elementlarini chiqaring;
2. Foydalanuvchidan buyurtma qabul qiluvchi dastur yozing. Mahsulotlar nomini birma-bir qabul qilib, yangi ro‘yxatga joylang;
3. e-bozor uchun mahsulotlar va ularning narhlari lug‘atini shakllantiruvchi dastur yozing. Foydalanuvchidan lug‘atga bir nechta elementlar (mahsulot va uning narhi) kiritishni so‘rang.

3.2-§. Kortejlar.

Reja:

1. *Kortej va uning vazifalari;*
2. *Kortej elementlari ustida bajariladigan funksiyalar va ularning vazifalari.*

Tayanch so‘zlar: *kortej, tuple, list.*

Kortejlar yoki to‘plam (inglizcha: tuples) — Python dasturlash tilida ma’lumotlarni saqlash va ishlatish uchun mo‘ljallangan o‘zgarmas (immutable) ma’lumotlar strukturasi. Ular ichki o‘zgaruvchilarni bir joyda saqlashning eng samarali yo‘llaridan biri sifatida tanilgan. Kortejlar, ro‘yxatlar (list) bilan solishtirganda, o‘zgaruvchan emasligining afzalligi tufayli ko‘proq xavfsizlik va tezlikni ta’minlaydi.

Kortejlarni yaratish uchun oddiygina qavslar ichida elementlarni joylashtiramiz, masalan: my_tuple = (1, 2, 3). Kortejlardagi elementlar har xil turdagi ma’lumotlarni — raqamlar, matnlar yoki hatto boshqa

kolleksiyalarni o‘z ichiga olishi mumkin. Bu esa ularni dasturlashda juda qulay va foydali hisoblanadi.

Kortejlar ko‘plab amaliy qo‘llanilishlarga ega. Masalan, ularni funksiyalarga ko‘p qiymatlarni qaytarish uchun ishlatish mumkin. Shuningdek, parallellikni saqlash va ma’lumotlar to‘plamini oson boshqarish uchun juda samarali vosita hisoblanadi.

Umuman olganda, kortejlar Python dasturlash tilida kuchli va samarali ma’lumotlar tuzilmasidir, bu esa dasturchilar uchun ko‘plab imkoniyatlar ochadi. Kortejlar yordamida ma’lumotlarni tartibga solish va ularni bir joyda saqlash jarayonini yanada osonlashtiradi.

Ro‘yxatlarni **tuple()** funksiyasi orqali kortejlarga aylantirish imkoniyati mavjud. **List()** funksiyasi orqali esa kortejlarni ro‘yxatlarga aylantirish imkoniyati mavjud.

Python dasturlash tilida kortejlarni faollashtirish quyidagicha amalga oshiriladi.

Kortej_o‘zgaruvchisi=(x1,x2,x3,…)

Kortejlardan foydalanishni quyidagi dastur orqali ko‘rib chiqamiz:

```
>>> a=('Ali', 'matematika', 'fanidan')
>>> a
('Ali', 'matematika', 'fanidan')
>>> b=(5, 'bahoga', "o'qiydi")
>>> b
(5, 'bahoga', "o'qiydi")
>>> a+b
('Ali', 'matematika', 'fanidan', 5, 'bahoga', "o'qiydi")
>>>
```

Yuqoridagi dastur kodidan foydalanib shuni aytishimiz mumkinki:

1. Kortej elementlari turli xil bo‘lishi mumkin;
2. Kortej elementlari ( ) qavs ichida vergul bilan ajratilib yoziladi;
3. Kortej elementlari ichida sitring turdagi ma’lumotlar kelsa tirnoq ichida yoziladi;

```
>>> a=(1,2,3,5,5,6)
>>> a
(1, 2, 3, 5, 5, 6)
>>> b=list(a)
>>> b
[1, 2, 3, 5, 5, 6]
>>> b[3]=4
>>> b
[1, 2, 3, 4, 5, 6]
>>> tuple(b)
(1, 2, 3, 4, 5, 6)
>>>
```

Kortej elementlarini o‘zgartirish uchun, kortejni list, ya’ni ro‘yxat ko‘rinishiga o‘tkazib olish talab etiladi, so‘ngra ro‘yxat elementlarini indeks raqamiga murojaat qilinib almashtiriladi.

Nazariy savollar

1. Kortej va uning vazifasi;
2. Kortej elementlari ustida qanday amallar bajariladi?
3. List() funksiyasi vazifasi?
4. Tuple() funksiyasi vazifasi?

Mustaqil ishlash uchun topshiriqlar

1. Natural n ta elementdan tashkil topgan kortej hosil qiling.
2. Butun n hamda haqiqiy sonli A(1:n) kortej hosil qiling. Uning elementlari uzunligini toping.
3. Butun n hamda haqiqiy sonli A(1:n) kortej hosil qiling. Uning eng katta va eng kichik elementlari orasidagi farqni toping.
4. Butun n hamda haqiqiy sonli A(1:n) kortej hosil qiling. Uning elementlari qatoriga yana bir haqiqiy sonni qo‘shing.
5. Haqiqiy x1, x2 va x3 sonlari berilgan. Butun sonli A(1:3) kortej elementlarini yi=xi-3xi formula yordamida aniqlang. Bu jadvalning barcha elementlari ko‘paytmasini hisoblang.

3.3-§. Python dasturlash tilida funksiyalar.

Reja:

1. *Funksiya haqida tushuncha;*
2. *Funksiya tanasini faollashtirish;*
3. *Funksiyaga bir nechta argument uzatish.*

Kalit so‘zlar: *funksiya, argument, def, print, range.*

Funksiya haqida tushuncha

Python dasturlash tili tarkibida dastur tuzish vaqtida ma’lum bir jarayonlar bir necha marta bajarilishi yoki bir necha marta murojat qilinishi mumkin. Dasturlash tillari tarkibida bir necha marta bajarilish kerak bo‘lgan jarayonlarni bir marta tasvirlab, keyin shu dasturga murojaat qilish imkoniyati mavjud. Dasturlash tillari tarkibida bir necha

marta bajarilishi mumkin bo‘lgan holatlarni qism dastur sifatida e’lon qilish va kerakli joyga shu qism dasturga murojat qilish mumkin.

Funksiya ma’lum bir vazifani bajarishga mo‘ljallangan kodlar yig‘indisi. Biz shu paytgacha bir nechta tayyor funksiyalardan foydalanib keldik. Misol uchun print() funksiyasi konsolga matn chiqarish uchun, range() funksiyasi esa ma’lum oraliqdagi sonlarni yaratish uchun ishlatiladi.

Aslida har qanday funksiyaning ortida ham bir necha qatordan iborat kod bo‘ladi, lekin biz funksiyaga murojat qilganda uning nomini yozamiz xolos. Funksiya ortidagi kod esa biz uchun yashirin bo‘lib qolaveradi. Funksiyalarning qulayligi ham shunda. Dastur davomida ma’lum bir kodlarni qayta-qayta yozmaslik uchun biz ularni jamlab, bitta funksiya ichiga joylashimiz va dastur davomida bu kodlarga funksiya nomi orqali murojat qilishimiz mumkin.

Ta’rif: *Dasturlash tilida tuzilgan dastur tarkibidagi ma’lum bir vazifalarni bajaruvchi qism dasturlar funksiyalar deyiladi.*

Funksiya tanasini faollashtirish.

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 nomi funksiya bajaradigan vazifadan kelib chiqqan holda qo‘yilishi maqsadga muvofiq, chunki dastur tuzuvchi

mutaxassislar uchun umumiy holda tushunish oson bo‘lishi kerak. Masalan, ketma-ketliklarni yig‘indisini hisoblash funksiyasiga **sum(),** to‘rtinchi darajali ildizni hisoblash uchun **sqrt4(),** sonning ekubini hisoblash uchun **ekub(),** sonning ekukini hisoblash uchun **ekuk()** va faktorialni hisoblash uchun **fakt()** deb nomlash maqsadga muvofiq bo‘ladi. Funksiya tarkibiga kiritilishi kerak bo‘lagan o‘zgaruvchi argumentlar, albatta, qavs ichida vergul bilan ajratilib yozilishi kerak[3].

def<funksiya_nomi>(<argumentlar>):

funksiya tanasi

natija

Keling oddiy, salom_ber deb nomlangan funksiya yaratamiz. Bu funksiyaga murojaat qilganimizda konsolga "Assalom alaykum!" degan xabarni chiqarsin.

def salom_ber(): '"Salom beruvchi funksiya’" print('Assalomu alaykum') salom_ber()	Natija: Assalomu alaykum

Funksiyaga nom berishda fe’l, ya’ni harakatni bildiruvchi so‘zlar yoki jumlalardan foydalaning. Bu bilan siz o‘zgaruvchi va funksiya o‘rtasini farqlashingiz oson bo‘ladi. Misol uchun, yuqorida biz funksiyamizni salom emas salom_ber deb nomladik.

Avvalgi sodda funksiyamiz foydalanivchidan hech qanday qiymat olmaydi va barchaga birday "Assalomu alaykum!" deb javob qiladi. Keling funksiyaga o‘zgartirish kiritamiz, funksiya foydalanuvchi ismini

qabul qilib, unga ismi bilan murojat qilsin. Buning uchun funksiya nomidan keyin, qavs ichida foydalanuvchi berishi kerak bo‘lgan qiymatni ko‘rsatamiz.

```
def salom_ber(ism):
    '"Fodyalanuvchi ismini qabul qilib, unga salom beruvchi funksiya’"
    print(f" Assalomu alaykum hurmatli {ism.title()}")
salom_ber('hasan')
```

Natija

```
RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python38-32/funksiya.py
 Assalomu alaykum hurmatli Hasan
```

Agar funksiyaga murojat qilishda, unga qiymat bermasak xatolik vujudga keladi:

salom_ber()

Natija: TypeError: salom_ber() missing 1 required positional argument: 'ism'

Funksiyaga bir necha bor murojat qilish. Funksiya yaratishning asl maqsadlaridan biri, biz unga qayta-qayta, yangi qiymatlar bilan murojat qilishimiz mumkin

```
def salom_ber(ism):
    '"Fodyalanuvchi ismini qabul qilib, unga salom beruvchi funksiya’"
    print(f" Assalomu alaykum hurmatli {ism.title()}")
salom_ber('hasan')
salom_ber('olim')
salom_ber('ali')
```

Natija

```
Assalomu alaykum hurmatli Hasan
Assalomu alaykum hurmatli Olim
Assalomu alaykum hurmatli Ali
```

Funksiyaga bir nechta argument uzatish

Shunday funksiyalar bor, bir emas bir nechta parameter talab qilishi mumkin, foydalanuvchi esa o‘z navbatida bir nechta argumentlar taqdim qilishi kerak. Funksiyaga argument uzatishning bir necha usuli bor, keling ular bilan birma bir tanishamiz.

To‘g‘ri tartibda uzatish. Bu usulda, funksiya parametrlari qaysi tartibda yozilgan bo‘lsa, argumentlar ham aynan shu ketma-ketlikda uzatilishi kerak. Keling bitta misol ko‘ramiz. Quyidagi funksiya foydalanuvchining ismi va familiyasini parametr sifatida qabul qilib, ularni jamlab xabar chiqaradi.

```
def toliq_ism(ism, familiya):
    """Foydalanuvchi ism va familiyasini jamlab chiqaruvchi funksiya"""
    print(f"Foydalanuvchi ismi: {ism.title()}\n"
          f"Foydalanuvchi familiyasi: {familiya.title()}")
toliq_ism('hasan', 'olimov')
```

Yuqoridagi funksiya to‘g‘ri natija chiqarishi uchun argumentlarni ism va familiya ketma-ketligida kiritishimiz lozim.

Natija

```
Foydalanuvchi ismi: Hasan
Foydalanuvchi familiyasi: Olimov
```

Agar argumentlarni noto‘g‘ri ketma-ketlikda bersak, natija ham biz kutganday chiqmaydi:

```
toliq_ism('hakimov','olim')
Natija:
Foydalanuvchi ismi: Hakimov
Foydalanuvchi familiyasi: Olim
```

```
def yosh_hisobla(ism, tugilgan_yil):
    """Foydalanuvchi yoshini hisoblaydigan dastur"""
    print(f"{ism.title()} {2024-tugilgan_yil} yoshda")
yosh_hisobla('muhriddin', 1996)
```

Natija

```
= RESTART:
C:/Users/Admin/AppData/Local/Programs/Python/Python38-
32/funksiya.py
Muhriddin 28 yoshda
>>>
```

Yuqoridagi kabi holatlarning oldini olish uchun argumentlarni parametr nomi bilan qo‘shib uzatishimiz mumkin. Buning uchun funksiyaga o‘zgartirish kiritish talab qilinmaydi.

```
def yosh_hisobla(ism, tugilgan_yil):
    """Foydalanuvchi yoshini hisoblaydigan dastur"""
    print(f"{ism.title()} {2024-tugilgan_yil} yoshda")
yosh_hisobla(tugilgan_yil=1996, ism='muhriddin')
```

Natija

```
= RESTART:
C:/Users/Admin/AppData/Local/Programs/Python/Python38-
32/funksiya.py
Muhriddin 28 yoshda
>>>
```

Nazariy savollar

1. Funksiya nima?
2. Funksiyaning vazifasi nima?
3. Funksiya tanasini faollashtirish qanday amalga oshiriladi?
4. Funksiyalarni shakillashtirishni umumiy ko‘rinishi?
5. Funksiyaga argument berishni soddalashtirish jarayonini tushuntirib bering.

Mustaqil ishlash uchun topshiriqlar

1. Foydalanuvchi ismi va yoshini so‘rab, uning tug‘ilgan yilini hisoblaydigan funksiya yozing.
2. Foydalanuvchidan son olib, uning kvadrati va kubini konsolga chiqaruvchi funksiya yozing.
3. Foydalanuvchidan son olib, son juft yoki toqligini konsolga chiqaruvchi funksiya yozing..
4. Foydalanuvchidan ikkita son olib, ulardan kattasini konsolga chiqaruvchi funksiya yozing. Agar sonlar teng bo‘lsa "Sonlar teng" degan xabarni chiqaring.

5. Kiritilgan A haqiqiy musbat sonining chap tarafiga (boshiga) B raqamini (1<=B<=10) qo‘shiluvchi funksiya hosil qiling.

6. Natural sonning raqamlari soni va raqamlari yiig‘indisini hisoblovchi funksiya hosil qiling.

3.4-§. Fayllar ustida amallar bajarish.

Reja:

1. *Fayllar haqida umumiy tushuncha.*
2. *Fayllarni faollashtirish.*
3. *Fayllar ustida amallar bajarish.*

Tayanch so‘zlar: *fayl, open, write, read, del, mantiqiy nom, fizik nom.*

Python dasturlash tilida fayllar bilan ishlash juda muhim va qulay jarayon. Bu jarayon dasturchilarga ma’lumotlarni saqlash, o‘qish va tahrirlash imkonini beradi. Pythonning asosiy funksiyalari yordamida fayllar ustida bajariladigan amallarni osonlik bilan amalga oshirish mumkin.

Ta’rif: Kompyuterning xotirasida alohida nom bilan saqlanadigan ma’lum bir turga mansub ma’lumotlar majmuasi fayl deb nomlanadi.

Python dasturlash tilida fayl nomi ikki xil bo‘ladi

1. Mantiqiy nom
2. Fizik nom

Faylning fizik nomi kompyuter xotirasida .txt kengaytmali nom bilan saqlangan fayl hisoblansa, faylning mantiqiy nomi dastur tuzish jarayonida biriktirilgan o‘zgaruvchi hisoblanadi.

Python dasturlash tilining yana bir imkoniyatlaridan biri bu fayllar bilan ishlashdir. Python dasturlash tili orqali fayllar ustida quyidagi amallarni bajarish mumkin. Yaratish, o‘qish, yangilash va o‘chirish.

Fayllar bilan ishlash

Fayllar bilan ishlashda asosiy funksiyasi sifatida **open()** funksiyasi ishlatiladi.

Fayllarni ochishning usullari

1. **“r”**- O‘qish - Standart qiymat. Faylni o‘qish uchun ochadi, agar fayl mavjud bo‘lmasa, xatolik beradi.
2. **“a”**- Qo‘shish - faylni qo‘shish uchun ochadi, agar u mavjud bo‘lmasa, uni yaratadi
3. **“w”**- Write - faylni yozish uchun ochadi, agar u mavjud bo‘lmasa, uni yaratadi
4. **“x”**- Yaratish - Belgilangan faylni yaratadi, agar fayl mavjud bo‘lsa, xatoni qaytaradi

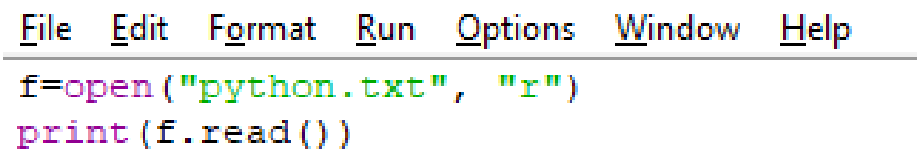
Agar siz faylni hech qanday rejimda belgilamasdan ochmoqchi bo‘lsangiz u holda fayl nomini o‘zini yozib o‘tasiz. bunda odatiy "r" ya’ni faylni ochish uchun bo‘ladi.

```
f = open("fayl_nomi.txt")
```

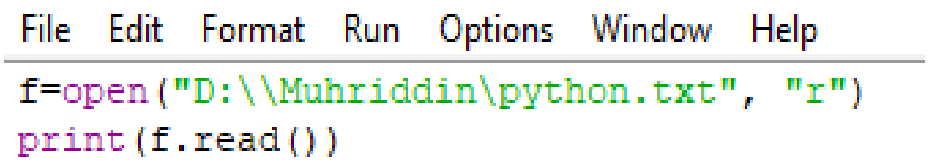
Python Fayl o‘qish

Fayllarni ochish uchun open() funksiyasidan foydalaning. Faylni o‘qish uchun read() funksiyasidan foydalaniladi. quyidagi misolda read

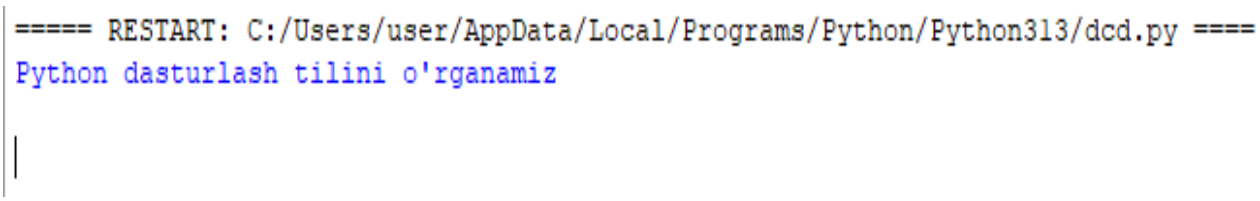
funksiyasi open() ga yo‘naltirilgan holda ishlaydi(doim shunday holat bo‘ladi).



Agar o‘qilayotgan fayl boshqa joyda bo‘lsa fayl_nomi oldidan uning joylashgan joyi ko‘rsatilib o‘tilishi zarur.



Natija:



Fayl n - belgigacha o‘qish.

Faylni ma’lum bir belgigacha o‘qish uchun read() funksiyasiga belgi sonini junatishimizni o‘zi kifoya qiladi.

f=open("D:\\Muhriddin\python.txt", "r")

print(f.read(3))

natija:

```
===== RESTART: C:/Users/user/AppData/Local/Programs/Python/Python313/dcd.py ====
Pyt
```

Fayllarni yopish

Ishni tugatgandan so‘ng, faylni albatta yopish kerak bo‘ladi chunki ochiq turgan faylning ma’lumotlariga zarar yetishi mumkin. Buning uchun close() funksiyasidan foydalanamiz.

File Edit Format Run Options Window Help

```
f=open("D:\\Muhriddin\python.txt", "r")
print(f.read())
f.close()
```

Pythonda faylga yozish va yaratish.

Mavjud faylga yozing Mavjud faylga yozish uchun **open()**funksiyaga parametr qo‘shishingiz kerak:

- **"a"**- Append - fayl oxiriga qo‘shiladi
- **"w"**- Write - har qanday mavjud tarkibni qayta yozadi

File Edit Format Run Options Window Help

```
f=open("D:\\Muhriddin\python.txt", "a")
f.write("qani ketdik!")
f.close()

f = open("D:\\Muhriddin\python.txt", "r")
print(f.read())
```

Natija:

```
===== RESTART: C:/Users/user/AppData/Local/P
Python dasturlash tilini o'rganamiz
qani ketdik!
```

Yangi fayl yaratish

Python dasturida yangi fayl yaratish uchun **open()** funksiyasi quyidagi parametrlarning biridan foydalaning:

- **"x"**- Yaratish - fayl yaratadi, agar fayl mavjud bo‘lsa, xatoni qaytaradi
- **"a"**- Qo‘shish - ko‘rsatilgan fayl mavjud bo‘lmasa fayl yaratadi
- **"w"**- Write - ko‘rsatilgan fayl mavjud bo‘lmasa fayl yaratadi

Yangi_fay.txt faylni yaratish

```
File  Edit  Format  Run  Options  Window  Help
f=open("yangi_fayl.txt", 'x')
```

Yangi bo‘sh fayl yaratildi.

Pythonda faylni o‘chirish.

Python dasturlash tilida faylni o‘chirish uchun **os** modulidan foydalanimiz. Os modulini ishlatish uchun **os.remove()** funksiyasidan foydalanish kerak.

Masalan:

import os

os.remove("yangi_fayl.txt")

Ushbu dastur kodi natijasida yuqorida yaratilgan yanfi_fayl nomi faylimiz o‘chib ketdi.

Mavjud faylni tekshirish.

Xato va kamchiliklarga yo‘l qo‘ymaslik uchun faylni o‘chirishdan oldin uning mavjudligini tekshirishimiz mumkin.

Masalan:

```
File  Edit  Format  Run  Options  Window  Help
import os
if os.path.exists('yangi_fayl.txt'):
    os.remove("yangi_fayl.txt")
else:
    print("fayl topilmadi")
```

Natija:

```
>>
===== RESTART: C:/Users/user/AppData/Local/Programs/Python/Python313/dcd.py ====
fayl topilmadi
>>
```

Papkani o‘chirish.

Butun jildni o‘chirish uchun os.rmdir() funksiyasidan foydalaniladi.

Import os

Os.rmdir(“mening_papkam”)

Nazariy savollar.

1. Fayllar haqida umumiy tushuncha.
2. Fayllarni faollashtirish qanday amalga oshiriladi?
3. Fayllar ustida qanday amallar bajariladi?
4. Faylning fizik va mantiqiy nomi haqida tushuncha.
5. Append funksiyasi haqida tushuncha.

Mustaqil ishlash uchun topshiriqlar.

1. Yangi myfile.txt fayl yarating.
2. Yangi fayl yarating va dasturda for sikl orqali 1 dan 20 gacha bo‘lgan sonlardan 5 tasini joylashtiring.
3. Yaratilgan faylga python dasturi orqali murojaat qiling.
4. Faylni oching va ichiga “Algoritmik tillar va dasturlash asoslari” so‘zini kiriting.
5. Fayl ichidagi ma’lumotning 11 belgisigacha o‘qing.
6. a satr berilgan. Agar a fayl nomi bo‘lishi uchun to‘g’ri kelsa, shu nom bilan bo‘sh fay fayl hosil qilinsin va rost chop etilsin. Agar a nom bilan fayl hosil qilish mumkin bo‘lmasa yolg’on chop etilsin.
7. Haqiqiy sonli fay nomi berilgan bo‘lsin. Fayldagi elementlar soni topilsin.

3.5-§. Pythonda geometrik shakllar chizish, chiziq qalinligi va harakatlanish vaqti.

Reja:

1. *Pythonda geometrik shakillar chizish.*
2. *Chiziq qalinligi.*
3. *Harakatlanish vaqti.*

Kalit so‘zlar: *turtle, matplotlib, pensize(), speed(), numpy, math, screen(), left(), forward(), pygame(), color().*

Python dasturlash tilida geometrik shakllarni chizish, chiziq qalinligini o‘zgartirish va harakatlanish uchun bir nechta modullar mavjud. "Turtle" moduli orqali grafik chizish va animatsiyalar yaratish mumkin. Bundan tashqari, "matplotlib" va "pygame" kabi modullar ham grafik tashkil etish, chiziq qalinligini o‘zgartirish va animatsiyalar yaratish uchun ishlatiladi.

Python dasturlash tili tarkibida boshqa dasturlash tillari kabi grafik rejimi va uning imkoniyatlari mavjud. Chizmalar va sohalarni hosil qilish uchun python dasturlash tilida matplotlib kutubxonasini chaqirish kerak. Matplotlib kutubxonasini faollashtirishning umumiy ko‘rinishi quyidagicha.

from matplotlib.pyplot import*

“Turtle” - bu chizma taxtasi kabi Python -ning o‘ziga xos xususiyati bo‘lib, u bizga turtleni hamma joyini chizishni buyuradi! Turtle.forward (...) va turtle.right (...) kabi funktsiyalarni ishlatishimiz mumkin, ular Toshbaqa (Turtle)ni harakatga keltiradi. Turtlening keng tarqalgan usullari:

Usul	**Parametr**	**Tavsif**

Usul	Parametr	Tavsif
Turtle()	Hech nima	Yangi tutrle ob'ektini yaratadi va qaytaradi
forward()	miqdori	Toshbaqa (Turtle)ni belgilangan miqdorda oldinga siljitadi
backward()	miqdori	Toshbaqa (Turtle)ni belgilangan miqdorda orqaga siljitadi
right()	burchak	Toshbaqa (Turtle)ni soat yo‘nalishi bo‘yicha aylantiradi
left()	burchak	Toshbaqa (Turtle)ni soat sohasi farqli o‘girib
penup()	Hech nima	Toshbaqa (Turtle) qalamini oladi
pendown()	Hech nima	Toshbaqa (Turtle) qalamini qo‘yadi
up()	Hech nima	Toshbaqa (Turtle) qalamini oladi
down()	Hech nima	Toshbaqa (Turtle) qalamini qo‘yadi
color()	Rang nomi	Toshbaqa (Turtle) qalamining rangini o‘zgartiradi
fillcolor()	Rang nomi	Turtlening rangini o‘zgartirish ko‘pburchakni to‘ldirishda ishlatiladi
heading()	Hech nima	Joriy sarlavhani qaytaradi
position()	Hech nima	Joriy pozitsiyani qaytaradi
goto()	x, y	Toshbaqa (Turtle)ni x, y holatiga o‘tkazing
begin_fill()	Hech nima	To‘ldirilgan ko‘pburchak uchun boshlang'ich nuqtani eslang
end_fill()	Hech nima	Ko‘pburchakni yoping va joriy rang bilan

Usul	Parametr	Tavsif
		to‘ldiring
dot()	Hech nima	Nuqtani hozirgi holatida qoldiring
stamp()	Hech nima	Hozirgi joyda Toshbaqa (Turtle) shakli haqida taassurot qoldiradi
shape()	shakl nomi	"Ok", "klassik", "Toshbaqa (Turtle)" yoki "doira" bo‘lishi kerak

Turtle yordamida chizmachilik

Toshbaqa (Turtle) usullari va funktsiyalaridan foydalanish uchun biz Toshbaqa (Turtle)ni import qilishimiz kerak. "Toshbaqa (Turtle)" standart Python to‘plami bilan to‘ldirilgan va uni tashqaridan o‘rnatish shart emas. Turtle dasturini bajarish uchun yo‘l xaritasi 4 bosqichdan iborat:

Turtle modulini import qiling

1. Boshqarish uchun Toshbaqa (Turtle) yarating.
2. Toshbaqa (Turtle) usullaridan foydalanib, atrofni chizib oling.
3. done () ni ishga tushiring.

Yuqorida aytib o‘tilganidek, Toshbaqa (Turtle)ni ishlatishdan oldin, biz uni import qilishimiz kerak. Biz uni quyidagicha import qilamiz:

```
from turtle import *
# or
```

```
import turtle
```

Turtlelar kutubxonasini import qilib, Toshbaqa (Turtle)ning barcha funktsiyalarini bizga taqdim etgandan so‘ng, biz yangi chizma taxtasi (oyna) va Toshbaqa (Turtle)ni yaratishimiz kerak. Shunday qilib, biz kodni quyidagicha yozamiz:

```
wn = turtle.Screen()
wn.bgcolor("light green")
wn.title("Turtle")
shakl =Turtle()
```

Endi biz deraza va Toshbaqa (Turtle)ni yaratdik, Toshbaqa (Turtle)ni siljitishimiz kerak. Shakl qaragan tomonga 100 piksel oldinga siljish uchun biz kodlaymiz:

```
shakl.forward(100)
```

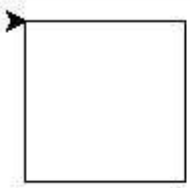
Biz shakl 100 piksel oldinga siljitdik, Ajoyib! Endi biz dasturni done () funktsiyasi bilan yakunlaymiz.

```
turtle.done ()
```

Shunday qilib, biz 100 piksel uzunlikdagi chiziq chizadigan dastur yaratdik. Turtle usullari yordamida biz har xil shakllarni chizishimiz va turli ranglarni to‘ldirishimiz mumkin. Pythondagi Turtlelar kutubxonasi yordamida kodlash uchun ko‘plab funktsiyalar va dasturlar mavjud. Keling, ba’zi asosiy shakllarni chizishni o‘rganamiz.

1 -shakl: kvadrat

```
from turtle import *

shakl=Turtle()
for i in range(4):
    shakl.forward(80)
    shakl.right(90)
turtle.done()
```

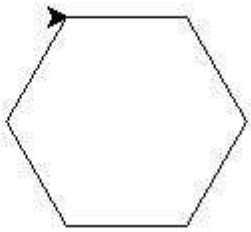


2 -shakl: yulduzcha

```
from turtle import *

yulduz=Turtle()

yulduz.right(75)
yulduz.forward(100)
for i in range(4):
    yulduz.right(144)
    yulduz.forward(100)

turtle.done()
```

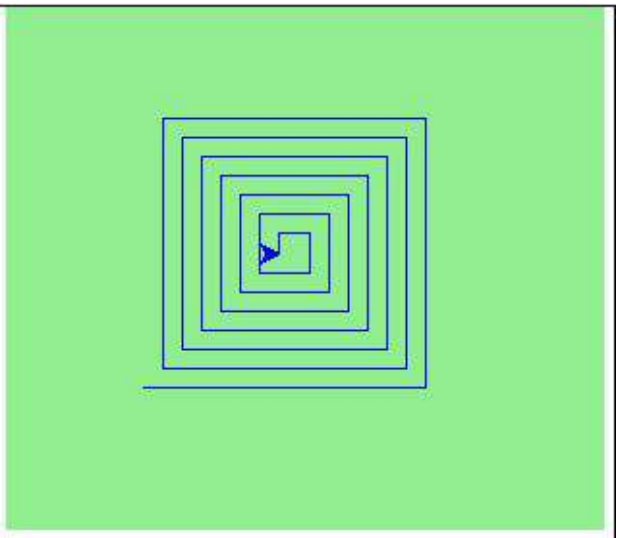
3 -shakl: olti burchakli

```
from turtle import *

oltiburchak=Turtle()

tomon=6
olti_b=60
burchak=360.0/tomon

for i in range(tomon):

    oltiburchak.forward(burchak)
    oltiburchak.right(olti_b)

turtle.done()
```



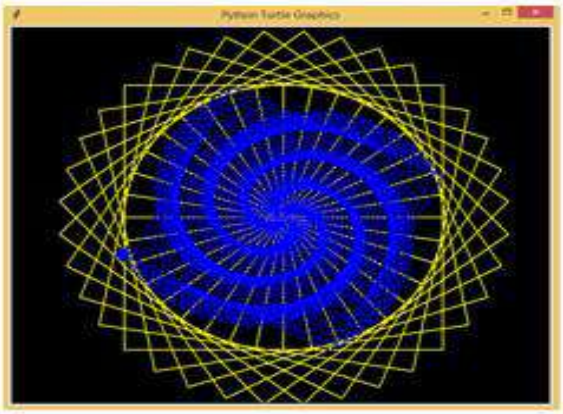
4-shakl. Ichma-ich spiral kvadrat

```
from turtle import *

wn = Screen()
wn.bgcolor("light green")
wn.title("Turtle")
skk = Turtle()
skk.color("blue")

def sqrfunc(size):
    for i in range(4):
        skk.fd(size)
        skk.left(90)
        size = size-5

sqrfunc(146)
sqrfunc(126)
sqrfunc(106)
sqrfunc(86)
sqrfunc(66)
sqrfunc(46)
sqrfunc(26)
turtle.done()
```



5-shakl. Pythonda sipiral chizma.

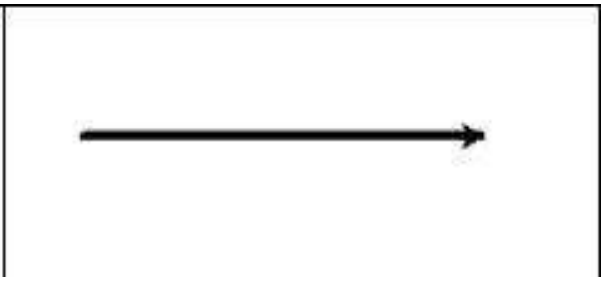
```
from turtle import Turtle, Screen

def draw_square(some_turtle):

    for _ in range(4):
        some_turtle.forward(200)
        some_turtle.right(90)

def draw_art():


    brad = Turtle(shape="turtle")
    brad.color("yellow")
    brad.pensize(2)
    brad.speed("normal")

    for _ in range(36):
        draw_square(brad)
        brad.right(10)

    # Turtle Angie
    angie = Turtle(shape="turtle")
    angie.color("blue")
    angie.pensize(2)
    angie.speed(5)
    size = 1

    for _ in range(300):
        angie.forward(size)
        angie.right(91)
        size += 1

window = Screen()
window.bgcolor("black")

draw_art()

window.exitonclick()
```



Активация Windows

6-shakl. Chiziq chizish.

```
from turtle import Turtle, Screen

tr = Turtle()
tr.pensize(4)
tr.forward(200)
turtle.done()
```



Turtle Moduli: Quyidagi misol orqali to‘rtburchak chizishni ko‘rish mumkin:

```
import turtle
t=turtle.Turtle()
for i in range(4):
    t.forward(200)
    t.left(90)
turtle.done()
```

Bu dasturda, "Turtle" modulining forward va left metodlari orqali to‘rtburchak chizadi.

Chiziq qalinligi

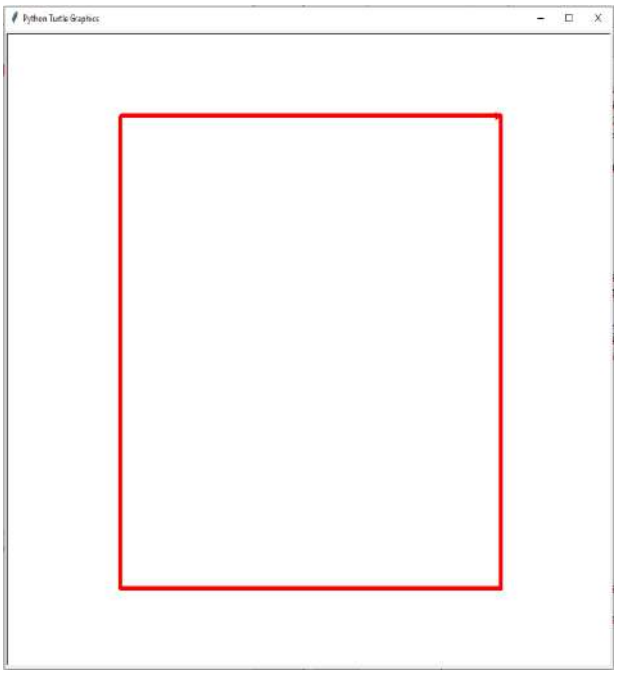
Pensize() bu metod orqali chiziq qalinligini belgilashingiz mumkun bo‘ladi. Pensize() metodini ishlatish quydagicha bo‘ladi

<o‘zgaruvchi>.pensize(qiymat)

Pensize metodi int yoki float turidagi ma’lumotlarni o‘z ichiga oladi

Quydagi misol orqali pensize() metodini ishlatib ko‘ramiz

```
from turtle import Turtle, Screen
oyna=Screen()
chiziq=Turtle()
chiziq.color('red')
chiziq.pensize(5.5)
chiziq.speed()
#chiziq.hideturtle()
chiziq.up()
chiziq.goto(300, 300)
chiziq.down()
chiziq.goto(300, -300)
chiziq.goto(-300, -300)
chiziq.goto(-300, 300)
chiziq.goto(300, 300)
oyna.mainloop()
```



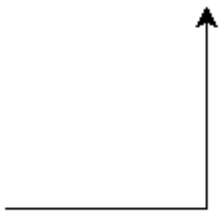
Harakatlanish vaqti

Harakatlanish vaqti **speed**(qiymat) metodi orqali amalga oshiriladi bu metod butun tipga mansub sonlarni qabul qiladi.

Speed() metodi harakatlanish tezligini amalga oshirib beradi, ya’ni belgilangan chizmani tezroq amalga oshirgani uchun harkatlanish vaqti deb ataymiz.

Speed() metodini quydagi misolda ko‘rishimiz mumkin:

```
from turtle import Turtle
a=Turtle()
a.forward(100)
a.left(90)
a.forward(100)
a.speed(1)
```



Dastur natijasi 100 qadam oldinga 90 gradus chapka va oldinga 100 qadam yurdi. Harakat tezligi 1 ga teng.

Nazariy savollar

1. Grafik muhitini faollashtirish tushintirib bering.
2. Chiziq turlari va ranglarini faollashtirishni tushintiri bering.
3. Turtle modulini tushuntirib bering.
4. Chiziq qalinligini tushuntirib bering.
5. Harakatlanish vaqtini tushuntirib bering.

Mustaqil ishlash uchun topshiriqlar

1. Python dasturida turtle modulidan foydalanib uchburchak chizing.
2. Python dasturida chizilgan uchburchakni chiziq qalinligini belgilang.
3. Chizmangizga rang bering.
4. To‘rtburchak chizing va to‘rtburcha ichiga harakatlanuvchi doirani. hosil qiling doira to‘rtburchak devorlariga urilganda qaytsin.
5. Rangi qizil chiziq chizing.
6. Rangi ko‘k va fon rangi yashil chiziq chizing.
7. Rangi sariq aylana chizing.
8. Rangi sariq va fon rangi ko‘k aylana chizing.
9. Rangi yashil to‘rtbrchak chizing.
10. Rangi yashil va fon rangi qora to‘rtburchak chizing.
11. Rangi qizil to‘rtburchak va yulduzcha chizing.
12. Rangi qizil va fon rangi qora to‘rtburchak va yulduzcha chizing.
13. Rangi sariq kvadrat chizing.

14. Rangi sari va fon rangi yashil kvadrat va elips chizing.
15. Fon rangi qizil Ellips chizing.
16. Fon rangi ko‘k Ellips va sakkiz birchak chizing.

Xulosa

Ushbu o‘quv qo‘llanma dasturlash asoslarini o‘zlashtirishda yordam berish uchun mo‘ljallangan bo‘lib, unda dasturlashning eng muhim tamoyillari, usullari va tavsiyalari keltiriladi. Dasturlashni o‘rganish nafaqat yangi imkoniyatlarni ochadi, balki mantiqiy fikrlash va muammolarni hal qilish qobiliyatini ham rivojlantiradi. O‘quv qo‘llanmaning maqsadi va ahamiyati dasturlash asoslarini o‘rganishga qaratilgan yo‘l xaritasini taqdim etishdan iborat. Ushbu qo‘llanma nafaqat yangi boshlovchilar uchun muhimdir, chunki u dasturlash dunyosining zamonaviy tendentsiyalarini va metodologiyalarini yoritadi. Ushbu o‘quv qo‘llanma orqali o‘quvchilar dasturlash tillari, algoritmlar, ma’lumotlar tuzilmalari va dasturlash printsiplarini mukammal o‘zlashtirish imkoniyatiga ega bo‘ladilar.

O‘quv qo‘llanmaning ahamiyati shundaki, u nafaqat bilim berish, balki o‘quvchilarning motivatsiyasini oshirish va dasturlash jarayonlarini qiziqarli va samarali qilishdir. O‘quvchilar o‘z maqsadlariga erishishda ishonchli va qo‘llab-quvvatlovchi manba sifatida ushbu qo‘llanmani ishlatishlari mumkin. O‘quv qo‘llanmasi nafaqat o‘quv jarayonini yengillashtiradi, balki dasturlash sohasida muvaffaqiyatga erishish uchun zarur bo‘lgan ko‘nikmalarni shakllantirishga yordam beradi.

Glossariy

Algoritm – masala yoki muommoni hal etish uchun beriladigan chekli sondagi bo‘yruqlar ketma ketligi.

Algoritm xossalari – algoritmni amalga oshiradigan beshta xususiyatlar majmuasi.

Algoritm turlari – algoritmning bajarilishiga ko‘ra uning turlari.

Chiziqli algoritmlar – Algoritmning shartlarsiz chiziqli bajarilishi.

Tarmoqlanuvchi algoritmlar - Algoritmning shartlar asosida bajarilishi.

Takrorlanuvchi algoritmlar - Algoritmning ma’lum bir qismi ikki va undan ortiq bajarilishi

Komplyator – mashina tiliga kodni to‘liq tarjima qiladi.

Interpretator – mashina tiliga kodni qism qilib tarjima qiladi.

IDLE rejimi – Python tilining ish rejimi.

Interaktiv - Python tilining qismiy faol ish rejimi.

Faylli dastur – alohida nom bilan saqlanuvchi tuzilgan dastur.

ENTER – faollashtirish tugmasi.

Restart - Python tilining natijasini ko‘rsatuvchi kalitli so‘z.

Int – butun tur.

Float – haqiqiy tur.

Str- satrli tur.

Import – kutubxonlarni chaqirishning kalitli so‘zi.

Math – Matematik kutubxona.

Standart funksiya – dasturlash tili tarkibidagi funksiyalar.

Min – minimum funksiya.

Max – maksimum funksiya.

Sum – summator funksiya

Type – turni aniqlovchi funksiya.

Help – ma’lumot beruvchi funksiya.

Operator – malum bir bo‘yruqli kalit so‘z.

Chiziqli dastur –chiziqli algorit asosida tuzilgan dastur

If - mantiqiy kalitli so‘z.

Else – aks holda qismi uchun kalitli so‘z.

Elif – aks holdani ichidagi shart qismi uchun kalitli so‘z. **For** -parametrli sikl operatorining kalitli so‘zi.

Sikl – takrorlanish.

While - shartli sikl operatorining kalitli so‘zi.

Qism dasturlar –dastur tarkibidagi kichik dastur

Funksiya – bir yoki ko‘p qiymat qaytaruvchi funksiya.

Global – butun dastur davomida faol hisoblanadi.

Lokal – dasturning ma’lum bir qismi faol hisoblanadi.

Argument – funksiya qabul qiladigan qiymat.

Def – funksiyani faollashtirish so‘zi.

Return –qiymat qaytarish kalitli so‘zi.

Rekursiya – o‘z o‘ziga murojaat.

Rekursiv funksiya - o‘z o‘ziga murojaat qiluvchi funksiya.

Ro‘yxat - Har xil turga mansub bo‘lgan yagona nom bilan saqlanuvchi tartiblangan ma’lumotlar majmuasi.

Kortej- o‘zgartirilmaydigan ro‘yxat.

Sorted – saralash funksiyasi.

Append – ma’lumot qo‘shish funksiyasi.

Del - ma’lumot o‘chirish funksiyasi.

Numpy – massiv kutubxonasi.

Fayl – alohida nom bilan saqlanuvchi ma’lumotlar majmuasi.

Open – faylni ochish bo‘yrug‘i.

Read – fayldan ma’lumot o‘qish bo‘yrug‘

Write – faylga ma’lumot yozish bo‘yrug‘i.

Matplotlib - grafik kutubxonasi.

Plot – grafik chizish funksiyasi.

Savefig – grafikli ma’lumotni saqlash bo‘yrug‘i

Ylabel – Oy koordinatalar sistemasiga matn joylashtirish.

Title – koordinatalar sistemasiga matn joylashtirish.

Color –grafikga rang berish.

Foydalanilgan adabiyotlar ro‘yxati

Asosiy adabiyotlar.

1. M.M.Aripov, N.A.Otaxanov. Dasturlash asoslari. o‘quv qo‘llanma. “Tafakkur bo‘stoni” Toshkent-2015. 52-b
2. Axatov.A, Nazarov.F. Python tilida dasturlash asoslari (I-qism). o‘quv qo‘llanma. Samarqand – 2020. 10-bet.
3. M.O‘.O‘ktamov. Dasturlash tillari II fanidan (o‘quv qo‘llanma). «Grand kondor print» Toshkent – 2024.
4. Sh.A. Mengliyev, O.A. Abdug‘aniev, S.Q. Shonazarov, D. SH. To‘rayev. Python dasturlash tili. Termiz-2021.
5. M.A.Bobojonova, H.Sh.Rustamov., Python dasturlash tilida masalalar va uning yechimlari. O‘quv qo‘llanma.
6. M.A.Bobojonova. Python dasturlash tili. O‘quv qo‘llanma. Durdona nashiryoti. Buxoro – 2023.
7. A.Ibragimov. Python asoslari. Ushbu qo‘llanma internet tarmog’idagi ma’lumotlar asosida yaratildi.
8. Normurodov Ch.B. Mengliyev Sh.A. PHP7 dasturlash tili - O‘quv qo‘llanma – Termiz: “Xamidi xususiy firmasi”, 2020, 218 bet.
9. Vasilev A. N. Python na primerax. Prakticheskiy kurs po programmirovaniyu. — SPb. Nauka i Texnika, 2016. — 432 st.
10. Dj. Forse, P. Bisseks, U. CHan - Django. Razrabotka veb-prilojeniy na Python, Simvol-Plyus , 2010. – 456 st
11. Eric Matthes Python Crash Course, a hands-on, Project-Based, Introduction to Programming, ISBN-13: 978-1-59327-603-4, 2016, 562

Qo‘shimcha adabiyotlar.

1. Mirziyoev SH.M. Erkin va farovon, demokratik O‘zbekiston davlatini birgalikda barpo etamiz. O‘zbekiston Respublikasi Prezidenti lavozimiga kirishish tantanali marosimiga bag‘ishlangan Oliy Majlis palatalarining qo‘shma majlisidagi nutq, Toshkent, 2016. 56-b.

2. Mirziyoev SH.M. Tanqidiy tahlil, qat’iy tartib-intizom va shaxsiy javobgarlik – har bir rahbar faoliyatining kundalik qoidasi bo‘lishi kerak. Mamlakatimizni 2016 yilda ijtimoiy-iqtisodiy rivojlantirishning asosiy yakunlari va 2017 yilga mo‘ljallangan iqtisodiy dasturning eng muhim ustuvor yo‘nalishlariga bag‘ishlangan Vazirlar Mahkamasining kengaytirilganmajlisidagi ma’ruza, 2017 yil 14 yanvar’ –Toshkent, O‘zbekiston, 2017. 104-b.

3. Mirziyoev SH.M. Qonun ustuvorligi va inson manfaatlarini ta’minlash-yurt taraqqiyoti va xalq farovonligining garovi. O‘zbekiston Respublikasi Konstitutsiyasi qabul qilinganining 24 yilligiga bag‘ishlangan tantanali marosimdagi ma’ruza. 2016 yil 7 dekabrь-Toshkent, O‘zbekiston, 2017. 48-b.

4. Mirziyoev SH.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Mazkur kitobdan O‘zbekiston Respublikasi Prezidenti SHavkat Mirziyoevning 2016 yil 1 noyabrdan 24 noyabrga qadar Qoraqalpog‘iston Respublikasi,viloyatlar va Toshkent shaxri saylovchilari vakillari bilan o‘tkazilgan saylovoldi uchrashuvlarida so‘zlagan nutqlari o‘rin olgan.-Toshkent, O‘zbekiston, 2017. 488-b.

Foydalanilgan internet resurslari

1. https://python.sariq.dev – Python dasturlash tilida dastur tuzish bo‘yicha amaliy va nazariy bilimlar keltirilgan.
2. http://cppstudio.com –Python tilida programmalash bo‘yicha namunalar izoxlari bilan keltirilgan
3. http://cplusplus.com – Python tilida mavjud konstruksiyalar ta’rifi, ishlatish namunalari bilan keltirilgan.
4. http://www.compteacher.ru/programming - dasturlash bo‘yicha video darsliklar mavjud.
5. http://www.intuit.ru – internet universitet, dasturlash bo‘yicha yozma va video ma’ruzalar o‘qish, test sinovlaridan o‘tish va sertifikat olish imkoniyati mavjud.
6. http://www.metanit.com – Python dasturlash tillari, Django texnologiyasi bo‘yicha ma’ruzalar, dasturlar namunasini olish mumkin.
7. http://www.ziyonet.uz – dasturlash asoslari bo‘yicha referatlar topish mumkin.
8. http://www.tami.uz/kitob.php – dasturlash bo‘yicha yozma va video ma’ruzalar o‘qish va kitobni yuklab olish imkoniyati mavjud.

Sh.B.Axmedova, D.E.Toshtemirov, M.A.Ahtamqulov

DASTURLASH ASOSLARI

(O‘QUV QO‘LLANMA)

Muharrir: O.Abdullayev
Dizayner: J.Qobilov

“OMADBEK PRINT NUMBER ONE” MCHJ
Andijon shahri, Bobur shoh ko`chasi, 39 a uy.
Nashriyot litsenziya raqami: № 156677.
2023-yil 7-noyabrda berilgan.
Tijoriy takliflar uchun: 99-906-90-71

Bosmaxonaga 2025 yil 22 noyabrda berildi.
Bosishga 2025 yil 26 noyabrda ruxsat etildi.
Bichimi 84x108 1/16. Hajmi 7,9 bosma taboq.
Times New Roman garniturasi,
ofset qog`ozi, ofset usulida chop etildi.
Buyurtma 877 Adadi 50 dona.

Ushbu kitobning hech bir qismidan mualliflarning yozma ruxsatisiz elektron yoxud bosma shaklda tijoriy maqsadlarda foydalanish uchun chop etish hamda internet, ijtimoiy tarmoqlar orqali tarqatish mumkin emas.
Mualliflik huquqlari qonun bilan himoya qilinadi.

“OMADBEK PRINT NUMBER ONE” MCHJ
bosmaxonasida chop etildi.
Andijon shahar Bobur shoh ko`chasi, 39 a uy.
O`zbekiston Respublikasi Prezidenti adminstratsiyasi huzuridagi Axborot va ommaviy kommunikatsiyalar agentligining **19.02.2025** dagi **627268** raqamli guvohnomasi.