

PYTHON DASTURLASH TILI VA UNING IMKONIYATLARI

Sharipova Soraxon Toxirjon qizi

Andijon davlat universiteti Amaliy matematika yo'nalishi 3-bosqich talabasi

Karimova Begoyim Iqboljon qizi

Andijon davlat universiteti Amaliy matematika yo'nalishi 3-bosqich talabasi

Annotatsiya: Ushbu maqolada Python dasturlash tili va uning imkoniyatlarini o'rganish va qo'llashning soddaligi, mukammal kutubxonalarning mavjudligi, Pythonda ma'lumot turlari, Pythonda o'zgaruvchilarni tasniflash, Pythonda xatoliklar bilan ishlash, Python dasturlash tili kutubxonasi haqida batafsil bayon etilgan.

Kalit so'zlar: dasturchi, Python, dasturiy ta'minot, ma'lumotlar bazasi, dasturlash tili, parametr, interpretatsiya, matematik operatorlar.

Kirish. Hozirgi jadal rivojlanish va turli jarayonlarni avtomatlashtirish hamda robotlashtirish davrida dasturlashni bilish va uni o'z ish jarayonida ishlata olish texnik va pedagogik yo'nalishda ta'lim olayotgan o'quvchi-talabalar uchun juda muhim deb hisoblanadi. Bu zamonaviy mutaxassislar uchun eng zaruriy talablardan biridir. Hozirgi kunda barcha muammolarimiz kaliti kompyuter, telefon va undagi dasturlarga bog'liq bo'lib qolyapti. Shu sababli dasturlash sohasiga bo'lgan talab ham ortib bormoqda. Python dasturlash tili samarador yuqori darajadagi ma'lumotlar tuzilmasini hamda oddiy, ammo samarador bo'lgan ob'yektga yo'naltirilgan dasturlash uslublarini taqdim etadi. Budan tashqari, bu til o'rganish uchun oson va shu bilan birga imkoniyatlari yuqori bo'lgan oz sonli dasturlash tillari jumlasiga kiradi va shu bilan birgalikda unda dasturlash jarayoni juda ham oddiy amalga oshiriladi. Python dasturlash tilining rasmiy sayti – www.python.org bo'lib, uning muallifi *Gvido van Rossum* hisoblanadi. Pythonning o'ziga xosligi esa uning oddiyligi, o'rganishga osonligi, soddagina sintaksisga egaligi va dasturlash jarayonini boshlash uchun qulay, erkin va ochiq kodlik dasturiy ta'minotga

egaligidir. Undan tashqari, o'z dasturingizni yozish davomida quyi darajadagi detallarni, misol uchun xotirani boshqarishni hisobga olishingizga hech qanday hojat qolmaydi. Bu dasturlash tili ko'plab platformalarda hech qanday o'zgartirishlarsiz ishlay oladi va u interpretatsiya qilinadigan tillar jumlasiga mansub.

Adabiyotlar tahlili va metodologiya. Python – bu o'rganishga oson va shu bilan birga imkoniyatlari yuqori bo'lgan oz sonlik zamonaviy dasturlash tillari qatoriga kiradi. Python yuqori darajadagi ma'lumotlar strukturasi va oddiy lekin samarador obyektga yo'naltirilgan dasturlash uslublarini taqdim etadi.

Pythonning o'ziga xosligi:

1. Oddiy, o'rganishga oson, sodda sintaksisga ega, dasturlashni boshlash uchun qulay, erkin va ochiq kodlik dasturiy ta'minot;
2. Dasturni yozish davomida quyi darajadagi detallarni, misol uchun xotirani boshqarishni hisobga olish shart emas;
3. Ko'plab platformalarda hech qanday o'zgartirishlarsiz ishlay oladi;
4. Interpretatsiya qilinadigan til;
5. Kengayishga moyil til. Agar dasturni biror joyini tezroq ishlashini xoxlasak shu qismni C yoki C++ dasturlash tillarida yozib keyin shu qismni python kodi orqali ishga tushirsa(chaqirsa) bo'ladi;
6. Juda ham ko'p xilma-xil kutubxonalarga ega;
7. xml/html fayllar bilan ishlash;
8. http so'rovlari bilan ishlash;
9. GUI (grafik interfeys);
10. Veb saytlarni yaratish;
11. FTP bilan ishlash;
12. Robot texnikada;
13. Matematik va ilmiy hisoblashlarni dasturlash.

Pythonni katta projeklarda ishlatish mumkin. Chunki, uni chegarasi yo'q, imkoniyati yuqori. Shuningdek, u sodda va universalligi bilan dasturlash tillari orasida eng yaxshisidir.

Pythonning matematik funksiyalar kutubxonasi trigonometrik hisoblashlar, sonli shakl almashtirishlar va sonli almashtirishlarni bajaradi. Trigonometrik funksiyalar argumentlari radianlarda beriladi, hamda graduslarni radianga va aksincha almashtiruvchi funksiyalar ham mavjud. Matematik operatorlar bilan bir qatorda Pythonda ko'p sonli matematik funksiyalar ham nazarda tutilgan.

Python tilida turli tipdagi o'zgaruvchilardan foydalanish mumkin, shu sababli, har bir tipdagi o'zgaruvchilar qanday tavsiflanishini bilish zarur. Python tilida bitta o'zgaruvchini dastur bajarilishi davomida satr yoki son sifatida ishlatish mumkin. Shu bilan birga Python tilida o'zgaruvchilar bilan ishlanganda oshkor ko'rsatilishi mumkin bo'lgan asosiy ma'lumotlar tiplari to'plami mavjud.

Butun (integer) sonlar – Sonning kasr bo'lmagan son bo'lib, ularda sonning asosi (10 lik), o'n oltilik (asosi 16-prefiksga ega) yoki sakkizlik (asosi 8- prefiksli) sanoq sistemalar ko'rsatiladi.

Siljuvchi vergulli (float) sonlar – sonning kompyuterda amalga oshiriladigan eksponentsiol yozuvi. Xuddi shuningdek, "ikkilangan aniqlikga" ega bo'lgan son ham mavjud.

Satr (string) – simvollar ketma – ketligidan iborat bo'lib, unda har bir simvol bir bayt o'lchamdan, toki maksimal uzunlik 216 gacha bo'lgan joyni egallaydi. Yakka qavslarga olingan satrlar literallar sifatida qaraladi, ayni paytda 28 ikkilangan qavslar ichidagi satrlar esa (maxsus belgilar, o'zgaruvchilarning qiymatlari va shu kabilar) sifatida talqin qilinadi.

Bul (boolean) tipi - Mantiqiy ifoda bo'lib, uning qiymati faqat rost (True) yoki yolg'on (False) dan iborat.

Kompleks (complex) tipi – Sonning birinchi argument sifatida haqiqiy qism, ikkinchi argument sifatida mavhum qism uzatiladi.

Massiv (array) – bir nechta qiymatlarning tartiblashtirilgan xaritasi bo‘lib, undagi kalitlar qiymatlarga mos keladi. Kalitlar – bu indeks nomerlari (ular so‘zsiz tushuniladi) yoki aniq ko‘rsatilgan nishonlar.

Obyekt – bu berilganlarning xossalarini saqlovchi va berilganlarni qayta ishlash metodlaridan iborat bo‘lgan sinf.

Resurs – tashqi resursga havola bo‘lib, ular maxsus funksiyalar tomonidan yaratiladi va saqlanadi.

NULL – qiymatga ega bo‘lmagan o‘zgaruvchi. Bu o‘zgaruvchi, shakllantirilmagan bo‘ladi (unga hech bir qiymat berilmagan bo‘ladi), agar unga NULL o‘zgarmasi ta’minlangan bo‘lsa yoki unset() funksiyasi yordamida bekor qilinmagan bo‘lsa.

Asosiy natijalar. Arifmetik amallar va qiymat berish operatori. Berilganlarni qayta ishlash uchun Python tilida amallarning juda keng majmuasi aniqlangan. Amal - bu qandaydir harakat bo‘lib, u bitta (unar) yoki ikkita (binar) operandlar ustida bajariladi, hisob natijasi uning qaytariluvchi qiymati hisoblanadi. Tayanch arifmetik amallarga qo‘shish (+), ayirish (-), ko‘paytirish (*), bo‘lish (/), darajaga ko‘tarish (**), va bo‘lish qoldig‘ini olish (%) amallarini keltirish mumkin. Amallar qaytaradigan qiymatlarni o‘zlashtirish uchun qiymat berish amali (=) va uning turli modifikatsiyalari ishlatiladi: qo‘shish, qiymat berish bilan (+); ayirish, qiymat berish bilan (-); ko‘paytirish qiymat berish bilan (*); bo‘lish, qiymat berish bilan (/); bo‘lish qoldig‘ini olish, qiymat berish bilan (%) va boshqalar.

Odatda tabiat yoki jamiyatda uchraydigan turli muammo, masala yoki jarayonlarni o‘rganishni kompyuter yordamida olib borish uchun, birinchi navbatda, qaralayotgan masala, jarayon - obyektning matematik ifodasi, ya’ni matematik modelini ko‘rish kerak bo‘ladi. Qaralayotgan obyektning matematik modelini yaratish juda murakkab jarayon bo‘lib, o‘rganilayotgan obyektga bog‘liq ravishda turli soha mutaxassislarining ishtiroki talab etiladi. Umuman, biror masalani kompyuter yordamida yechishni quyidagi bosqichlarga ajratish mumkin. Qo‘yilgan chiziqli masalani kompyuterda yechish uchun, avval uning matematik modelini, keyin algoritmini va dasturini tuzish kerak bo‘ladi. Har

qanday murakkab algoritmnini ham uchta asosiy struktura yordamida tasvirlash mumkin. Bular ketma-ketlik, ayri va takrorlash strukturalaridir. Bu strukturalar asosida chiziqli, tarmoqlanuvchi va takrorlanuvchi hisoblash jarayonlarining algoritmlarini tuzish mumkin.

1-masala. Koordinatalari X_1, Y_1 va X_2, Y_2 ga teng bo'lgan nuqtalari orasidagi masofani hisoblang.

Yechish. Ikki nuqta orasidagi masofa $S = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ formula yordamida aniqlanadi.

```
import math
```

```
x1=float(input('x1 nuqta: '))
```

```
x2=float(input('x2 nuqta: '))
```

```
y1=float(input('y1 nuqta: '))
```

```
y2=float(input('y2 nuqta: '))
```

```
s=math.sqrt(math.pow((x2-x1),2)+math.pow((y2-y1),2));
```

```
print('Ikki nuqta orasidagi masofa= ',s)
```

Natija:

x1 nuqta:9

x2 nuqta:-6

y1 nuqta:-4

y2 nuqta:2

Ikki nuqta orasidagi masofa=16.15549442140351

2-masala. Birinchi hadi A , ayirmasi D , hadlari soni N ga teng arifmetik progressiyaning hadlarining yig'indisini hisoblang.

Yechish. Arifmetik progressiya istalgan hadi va hadlari yig'indisi uchun $a_n = a + d \cdot (n - 1)$, $S_n = \frac{2 \cdot a + d \cdot (n-1) \cdot n}{2}$, formulalar o'rinli bo'ladi.

```
a1=float(input('a1 birinchi hadi: '))
```

```
d=float(input('d ayirmasi: '))
```

```
n=float(input('n hadlari soni: '))
s=(2*a1+d*(n-1)*n)/2
print("Hadlari soni N ga teng arifmetik progressiya hadlarining yig`indisi= ",s)
```

Natija:

a1 birinchi hadi:8

d ayirmasi:2

n hadlari soni:10

Hadlari soni N ga teng arifmetik progressiya hadlarining yig`indisi=98.0

Kompleks sonni yaratish uchun $complex(a,b)$ funksiyasidan foydalanish mumkin. Bunda a - argument sifatida haqiqiy qism, b – argument sifatida, mavhum qismuzatiladi. Shuningdek, sonni $a + bj$ ko‘rinishida ifodalanadi.

3-masala. Kompleks son berilgan. Ushbu kompleks sonning mavhum qismini chiqaruvchi dastur tuzing.

```
a=int(input('a= '));
b=int(input('b= '));
s1=complex(a,b);
s2=s1.imag;
print(s1, 'kompleks sonning mavhum qismi= ',s2);
```

Natija:

a=1

b=-3

(1-3j) kompleks sonning mavhum qismi=-3.0

4-masala. Kompleks son berilgan. Ushbu kompleks sonning haqiqiy qismini chiqaruvchi dastur tuzing.

```
a=int(input('a= '));
b=int(input('b= '));
s1=complex(a,b);
s2=s1.real;
```

```
print(s1, 'kompleks sonning haqiqiy qismi= ',s2);
```

Natija:

a=1

b=-3

(1-3j) kompleks sonning haqiqiy qismi=1.0

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

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

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

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

Sodda funksiyaga misol.

```
def ikkita_sum():  
    a=int(input('a= '));  
    b=int(input('b= '));  
    sum=a+b;  
    return sum;
```

```
print(ikkita_sum());
```

Natija:

a=40

b=75

115

Bu misolda ikki son yig'indisini hisoblovchi funksiya ko'rsatilgan. Bu funksiya birorta argument qabul qilmaydi, yig'indini hisoblab natijani chiqaradi. Shundan so'ng **print** operatori tanasida natijani chiqarish uchun chaqiriladi.

Argument orqali uzatilgan qiymatni o'z ichiga oluvchi o'zgaruvchi, funksiya **parametri** deyiladi.

Xulosa. Python dasturlash tilining keng miqyosda qo'llanilishi mumkin bo'lgan uch asosiy soha bor: veb-dasturlash (backend – vebserver uchun ilovalar yozish), sun'iy intellekt masalalari, kompyuterda foydalanuvchi juda ko'p marta bajaradigan mayda ishlar (elektron xatlarni jo'natish, fayllarni izlash va bosmalash, elektron jadvaldan biror-bir ma'lumotlarni ajratib olish va xakozolar).

Python o'rganish ancha oson bo'lgan dasturiy tildir. Agar tabiiy tillar bilan o'xshatish qiladigan bo'lsak, biror-bir tilda fikrni yetkazish uchun ma'lum vaqt so'zlarni, tilning grammatikasi o'rganish kerak bo'ladi. Qandaydir minimal bilim shakllangandan so'ng, asta-sekin inson o'z fikrini ifoda eta boshlaydi. Dasturlash tillari bilan ham holat xuddi shunday. Biror dasturlash tilida amaliy foyda keltiradigan dastur yozishni boshlash uchun ma'lum bilimlar majmuini egallash kerak, shundan so'nggina dasturlashni boshlash mumkin. Boshqa dasturlash tillaridan farqli ravishda, Python da amaliy ahamiyatga ega dasturlarni ishlab chiqishga ancha ertaroq, hali tilning katta qismini o'rganmasdan turib ham kirishish mumkin.

FOYDALANILGAN ADABIYOTLAR:

1. Boltayev B., Azamatov A., Asqarov A., Sodiqov M., Azamatova G. Informatika va hisoblash texnikasi asoslari. Toshkent: “Cho‘lpon” nomidagi NMIU, 2015. – 160 b.
2. Kris Roffi. Kompyuter dasturlari. Python uchun dasturlash kitobi. – USA: Cambridge university press. 2017, – p. 204.
3. Vasilev A. N. Python na primerax. Prakticheskiy kurs po programmirovaniyu. — SPb. Nauka i Texnika, 2016. — 432 st.
4. Dj. Forse, P. Bisseks, U. CHan - Django. Razrabotka veb-prilojeniy na Python, Simvol-Plyus , 2010. – 456 st
5. Eric Matthes Python Crash Course, a hands-on, Project-Based, Introduction to Programming, ISBN-13: 978-1-59327-603-4, 2016, 562 p.
6. Dan Bader. Python tricks the book. Anja Pircher Design, 2017, – p. 299.